



Univerza v Mariboru

Fakulteta za naravoslovje
in matematiko



διάνοια

D I A N O I A

REVIJA ZA UPORABO NARAVOSLOVNO-MATEMATIČNIH ZNANOSTI

ISSN	2536-3565
Naslov publikacije/Title	DIANOIA , revija za uporabo naravoslovnih in matematičnih znanosti DIANOIA , journal for applications of natural and mathematical sciences
Letnik/Volume	9
Leto/Year	2025 (april)
Številka/Number	1
Založnik in izdajatelj/ Published & Issued by	Univerzitetna založba Univerze v Mariboru, Slomškov trg 15, 2000 Maribor, Slovenija, http://press.um.si/ , zalozba@um.si
Uredništvo/Editorial board	<i>odgovorni urednik/editor in chief</i> Mitja Slavinec <i>glavni urednik/executive editor</i> Drago Bokal <i>izvršna urednica/managing editor</i> Janja Jerebic <i>urednici za področje biologije/editors for biological sciences</i> Nina Šajna, Sonja Škornik <i>urednik za področje didaktike/editor for didactical sciences</i> Samo Repolusk <i>urednika za področje fizike/editors for physical sciences</i> Robert Repnik, Aleš Fajmut <i>urednika za področje matematike/editors for mathematical sciences</i> Igor Pesek, Janja Jerebic <i>urednik za področje tehnike/editor for technical sciences</i> Mateja Ploj Vrtič <i>tehnična urednica/technical editor</i> Špela Kajzer
Mednarodni uredniški svet/ International advisory board	Igor Emri (Fakulteta za strojništvo Univerze v Ljubljani, član SAZU), Matej Brešar (FNM, član SAZU), Sergey Pasechnik (Državna fakulteta v Moskvi), Vlad Popa-Nita (Fakulteta za fiziko Univerze v Bukarešti), Blaž Zmazek (FNM), Samo Kralj (FNM), Franci Janžekovič (FNM), Nataša Vaupotič (FNM), Mitja Kaligarič (FNM), Boris Aberšek (FNM), Andrej Šorgo (FNM), Bojan Mohar (Simon Fraser University, Vancouver), Matjaž Perc (FNM), Ivica Aviani (Naravoslovno matematična fakulteta Split), Fahriye Altınay (Univerza v Nikoziji), Andreas M. Hinz (Univerza Ludwig-Maximilians, München)
Oblikovanje/Design	Amadeja Bratuša
Lektoriranje/Proofreading	Ljudmila Bokal
Sedež uredništva/Address	FNM UM, Koroška cesta 160, 2000 Maribor
e-mail	dianoia@um.si
internet/web	www.fnm.um.si
Tisk/Printed by	FNM UM
Leto izida/Year	2025
Datum natisa/Published	2025
Naklada/Nr. of Copies	100 izvodov

Revija izhaja dvakrat letno, predvidoma aprila in septembra.

Kazalo / Table of Contents

Uvodnik

Drago Bokal 5

Spletni izzivi, ki vključujejo reševanje problemov s področja teorije grafov
Online Challenges That Include Solving Problems in Graph Theory

Jasmina Ferme, Daša Mesarič Štesl 9

Sinteza in karakterizacija kobaltove koordinacijske spojine s 4-cianopiridinom
Synthesis and Characterization of Cobalt Coordination Compound with 4-Cyanopyridine Ligand

Aljaž Čelofiga, Amalija Golobič, Brina Dojer 21

Tehnična analiza delnic: indikatorji in mere tveganja v programskem jeziku Python
Technical Analysis of Stocks: Indicators and Risk Measures in Python

Maša Galun, Mia Molnar, Ema Smolič, Timotej Jagrič, Dušan Fister 31

Optimizacija simulacije trgovanja z delnicami z večnitenjem v programskem jeziku Python
Optimization of Stock Trading Simulation with Multithreading in Python

Maša Galun, Mia Molnar, Ema Smolič, Drago Bokal 51

Crash Course on Hosting a Website with Focus on CORS
Hitri tečaj gostovanja spletne strani s poudarkom na CORS

Nikola Čokor, Jakob Battelino Prelog, Drago Bokal 67

Nova zlata doba naravoslovja in matematike ali družba med tragedijo in komedijo slehernikov

Drago Bokal

Univerza v Mariboru, Fakulteta za naravoslovje in matematiko, Koroška cesta 160, 2000 Maribor, Slovenija

Od filozofsko-etičnega preko povsem teoretičnega pa igrivo pustnega, fizikalnega, biološkega, ekonomskega, psihološkega do pravnega in še kakšnega modela poskušamo z zbirko kratkih esejev utemeljiti potrebo po iskanju resnice, zaupanju in hvaležnosti. Te koncepte predstavljamo kot podlago za vzporedne koncepte pravičnosti, demokratičnosti in poštenosti ter dualne koncepte inovativnosti, učinkovitosti in sprejetosti ter jih ilustriramo z omenjenimi modeli. Celoten razmislek utemeljuje etični imperativ, s katerim Trontelj dopolni Ricoeurjevo združitev Aristotela in Kanta: etični cilj je živeti dobro življenje z drugimi in za druge v pravičnem redu.

1 Filozofsko-etični matematični model

Svet je danes na razpotju. Bulletin of the Atomic Scientists [1], ki ga je leta 1945 ustanovil Albert Einstein skupaj s skupino znanstvenikov projekta Manhattan, je apokaliptično uro nastavil na 89 sekund do polnoči [2]. Zdi se, da je politika v kaosu interesov današnjega sveta izgubila kompas. Zdi se, da smo pozabili na Aristotela, ki je utemeljeval, da je "srednja pot, zlata pot" [3]. Vprašanje je, ali še verjamemo Kantu [4], ki je dve tisočletji kasneje navedeno nadgradil v etično vodilo "Ne stori drugemu, kar ne želiš, da drugi tebi storijo". Dobri dve stoletji in dve svetovni vojni kasneje je Ricoeur [5] oboje združil v "Etično vodilo je živeti dobro življenje z drugimi in za druge v pravičnih institucijah". Tudi to se zdi, da smo verjetno pozabili. Pred dobrim desetletjem je Trontelj [6] ob svoji izkušnji, da institucije lahko nehalo služiti človeštvu in postanejo same sebi namen, le-te nadomestil z nekoliko bolj abstraktnim redom in predlagal "Etično vodilo je živeti dobro življenje z drugimi in za druge v pravičnem redu".

So navedeno le pobožne želje idealističnih mislecev, ali pa je v poskusih omejevanja človeškega obnašanja, v pravilih, v odpovedovanju absolutni svobodi tudi kaka zakonitost? Kako blizu fizikalnim zakonitostim so lahko dogovorjena pravila? O tem bomo razmišljali v tem in naslednjih uvodnikih v reviji Dianoia.

Kakšen pa bo cilj teh razmislekov, bo abstraktno moraliziranje, pilatovsko umivanje rok in kazanje na drugega ali pa lahko v teh razmislekih iščemo kaj konkretnega, koristnega?

Vsekakor je smiselno zadnje. Da pa ta cilj zastavimo, uvedimo najprej dva pojma. Prvi je tragedija slehernikov [7]. V teoriji iger jo lahko čisto formalno definiramo. Poljudno

povedano pa je to igra v skupini, v kateri v povprečju posamezniki ustvarijo nekaj manj, kot porabijo. Taki igri bo zmanjkalo virov in se bo ustavila, privedla do revščine, boja za vire. Njeno nasprotje je komedija slehernikov [8]. Tam vsak prispeva nekaj več, kot iz igre odnese, z viška pa se lahko poskrbi za posameznike, ki ne zmorejo dovolj ustvariti, ali pa se s presežki razvija igro. Tragedija slehernikov se v skupnosti von Neumann-Morgensternovo racionalnih posameznikov zlahka zgodi sama od sebe. Za komedijo slehernikov pa je v prvi vrsti potrebno zaupanje. Nezaupanje v tragediji slehernikov in zaupanje v komediji slehernikov pa sta lahko tudi posledici sodelovanja v takih odnosih: posameznik, ki izhaja iz okolja, katerega odnosi ustrezajo komediji slehernikov, se lahko v tragediji slehernikov izkaže naiven; obratno lahko posameznik iz tragedije slehernikov v komediji slehernikov izstopa kot nezaupljiv.

Nazorna ilustracija obnašanja, ki izhaja iz tragedije slehernikov, je učinek opazovalca [9]. Leta 1964 je 38 ljudi v okrožju Queens v New Yorku od daleč pasivno opazovalo, kako neznanec zlorablja 28-letno natakario Catherine Genovese. Nihče ni priskočil na pomoč ali poklical policije, dokler je ni neznanec ubil in pobegnil. Majhen napor vsakega posameznega opazovalca bi lahko rešil življenje, a ker so bili navajeni boriti se vsak zase, nihče ni prevzel odgovornosti. Psihologi zato svetujejo, da če ste v stiski, ne vpijte "Na pomoč, na pomoč!" v upanju, da bo v bližini nek odgovoren dober samaritan, ki bo pomagal. Zavpite rajši "Gospod v rdečem suknjiču, prosim vas, pomagajte!", da se bo posameznik čutil nagovorjenega in bo prevzel odgovornost.

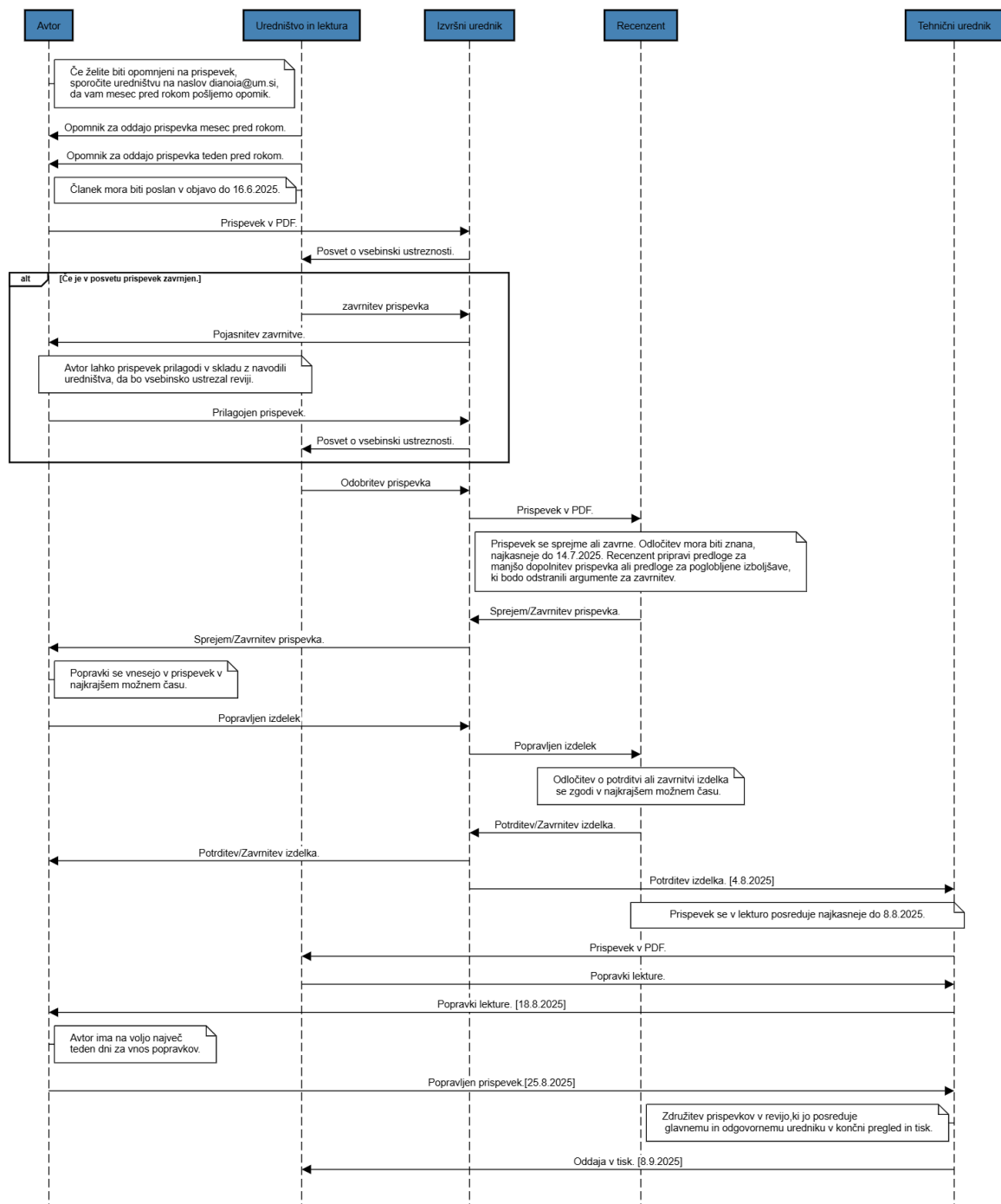
Pa je iz tragedije slehernikov mogoče priti v komedijo slehernikov? Tudi na to temo obstajajo znanstveni eksperimenti. Zapornikovo dilemo verjetno poznate [10]. Če ne, kratek opis igre: dva neodvisna igralca se morata odločiti ali za sodelovanje ali za tekmo. Če se oba odločita za sodelovanje, oba malo pridobita, če oba za tekmo, oba malo izgubita, v preostalem primeru pa veliko dobi tisti, ki se odloči za tekmo, drugi, ki se je odločil za sodelovanje, pa veliko izgubi. Če se igra ponovi samo enkrat ali pa je število njenih ponovitev znano, je njeno Nashevo ravnovesje tekma obeh igralcev in oba igralca izgubita. Če pa se igra ponavlja, pa se je eksperimentalno za optimalno izkazala strategija "kakor ti meni, tako jaz tebi". Na začetku ta strategija sodeluje, potem pa vrača to, kar je od soigralca prejela. V skladu s teorijo [11] mora imeti zmagovalna strategija naslednje lastnosti. Biti mora prijazna – začne s sodelovanjem (se spomnimo Trontlja, Ricoeurja - živeti dobro življenje, z drugimi in za druge?), maščevalna – soočena s tekmo neha sodelovati (tega prej omenjeni filozofi ne poudarjajo, na misel pa mi pride Darwin [12]), odpustljiva – soočena s sodelovanjem neha tekmovati (to predlaga Kant), in nezavistna – ne sme si prizadevati pridobiti več od drugih igralcev (o tem pa govori Aristotel). V določenih okoliščinah pa se izkaže, da je smiselno tudi, da je strategija nepredvidljiva – občasno, a ne prepogosto na tekmo reagira s poskusom sodelovanja, kar doda Axelrod [11].

Znamo najti prave priložnosti za tako sodelovanje?

Literatura

- [1] Bulletin of the Atomic Scientists. (n.d.). *The Bulletin*. Pridobljeno na: <https://thebulletin.org/>
- [2] Bulletin of the Atomic Scientists. (n.d.). *Doomsday Clock*. Pridobljeno na: <https://thebulletin.org/doomsday-clock/>
- [3] Aristoteles. (2002). *Nikomahova etika* (Reprint iz leta 1994, let. 38, str. 391). Slovenska matica.

- [4] Kant, I. (1993). Kritika praktičnega uma. *Filozofski vestnik*, 14(1), 201–231. <http://www.dlib.si/details/URN:NBN:SI:DOC-7UTK8T9Q>
- [5] Ricoeur, P. (1992). *Oneself as another*. University of Chicago press.
- [6] Trontelj, J. (2014). *Živeti z etiko* (1. izd., str. 316). Inštitut za etiko in vrednote Jože Trontelj.
- [7] Wikipedia avtorji. (n.d.). Tragedija skupnega. Pridobljeno na: https://en.wikipedia.org/wiki/Tragedy_of_the_commons
- [8] Rose, Carol (1986). "The Comedy of the Commons: Custom, Commerce, and Inherently Public Property". *The University of Chicago Law Review*. 53 (3): 711–781. doi:10.2307/1599583. ISSN 0041-9494. JSTOR 1599583.
- [9] Wikipedia avtorji. (n.d.). Učinek opazovalca. Pridobljeno na https://en.wikipedia.org/wiki/Bystander_effect
- [10] Wikipedia avtorji. (n.d.). Zapornikova dilema. Pridobljeno na https://en.wikipedia.org/wiki/Prisoner%27s_dilemma
- [11] Axelrod, R. (2006). *The Evolution of Cooperation* (Revised ed.). Basic Books. ISBN 0-465-00564-0.
- [12] Darwin, C. (2009). O nastanku vrst: z delovanjem naravnega odbiranja ali ohranjanje prednostnih ras v boju za preživetje (str. 421). Založba ZRC, ZRC SAZU.



Slika 1: Proces izdaje naslednje številke Dianoia. Vir:lasten

Spletni izzivi, ki vključujejo reševanje problemov s področja teorije grafov

Online Challenges That Include Solving Problems in Graph Theory

Jasmina Ferme, Daša Mesarič Štesl

Univerza v Mariboru, Pedagoška fakulteta, Koroška cesta 160, 2000 Maribor, Slovenija

Univerza v Mariboru, Fakulteta za naravoslovje in matematiko, Koroška cesta 160, 2000 Maribor, Slovenija

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija

Povzetek

Teorija grafov je ena izmed novejših vej matematike, zastopana na številnih področjih vsakdanjega življenja. Na osnovi konceptov s področja teorije grafov so med drugim nastale tudi številne igre, uganke, problemi in drugi izzivi, ki jih lahko igramo oziroma razrešujemo brez poznavanja grafov ter z njimi povezanih konceptov. Navedeno se velikokrat uporablja z namenom uresničevanja izobraževalnih ciljev ali krajšanja prostega časa. V tem prispevku navajamo nekaj primerov spletnih izzivov, pri katerih pravzaprav rešujemo probleme s področja teorije grafov, a za to znanja omenjene veje matematike ne potrebujemo. Poleg opisa izzivov predstavljamo tudi njihovo ozadje z vidika teorije grafov.

Ključne besede: teorija grafov, spletne igre, spletni izzivi.

Abstract

Graph theory is one of the newer branches of mathematics that is present in many areas of everyday life. Numerous games, puzzles, problems, and other challenges have been created based on concepts from graph theory and can be played or solved without knowledge of graphs and the concepts involved. These are often used to achieve educational goals or as free time activity. In this paper, we present some examples of online challenges where we actually solve problems from graph theory, but they do not require knowledge from this branch of mathematics. In addition to describing the challenges, we also present their background from the perspective of graph theory.

Key words: graph theory, online games, online challenges.

1 UVOD

Teorija grafov je ena izmed novejših, a zelo uveljavljenih matematičnih disciplin z izjemno širokim spektrom možnosti uporabe. Slednje izhaja iz dejstva, da lahko z grafi predstavimo ali opišemo marsikatero (realno) situacijo oziroma strukturo. Koncepte s področja teorije grafov

E-mail naslov/i: jasminda.firme1@um.si (Jasmina Ferme), Dasa.Stesl@fri.uni-lj.si (Daša Mesarič Štesl)

tako zasledimo na številnih področjih, na primer v kemiji, biologiji, računalništvu, ekonomiji, logistiki in sociologiji (več o uporabnosti konceptov s področja teorije grafov na različnih področjih lahko preberete na primer v [2, 6]).

Na osnovi teorije grafov ter z njimi povezanimi koncepti so nastale tudi številne igre, uganke, problemi in drugi izzivi, ki jih lahko igramo oziroma razrešujemo tudi brez poznavanja te matematične discipline. Omenjeno je včasih uporabljeno v izobraževalne namene (naj omenimo, da je teorija grafov na različne načine vključena tudi v izobraževanju otrok in dijakov, o čemer lahko več preberete v članku *Teorija grafov v osnovnošolskem in srednješolskem izobraževanju* [5]), včasih pa kot način, ki predstavlja prostočasno dejavnost. Zaradi nenehnega in hitrega razvoja digitalne tehnologije ter s tem sprememb v naših življenjskih slogih v zadnjih letih svojo razsežnost povečujejo predvsem spletne različice omenjenih izzivov.

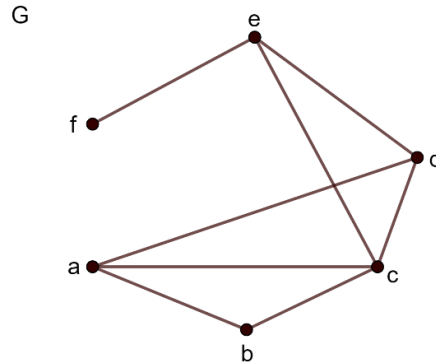
V tem prispevku bomo predstavili nekaj spletnih izzivov (iger, ugank, problemov, nalog ali podobnega), pri izvajanju katerih razrešujemo naloge oziroma probleme s področja teorije grafov. V večini bo šlo za spletne izzive, ki so uporabnikom znani pod izrazom *spletne igre*, pri čemer pa se izraz *igra* ne nanaša na klasično igro (aktivnost z vsaj dva igralcema, pri kateri dobimo na koncu zmagovalca ali neodločen rezultat), temveč na različne spletne izzive, uganke, naloge ali probleme, ki jih mora uporabnik razrešiti, da lahko napreduje na naslednje ravni izziva. Vsi predstavljeni izzivi bodo takšni, da jih lahko izvajamo brez poznavanja grafov in povezanih konceptov. Vsak izbran tip izziva bomo opisali, navedli nekaj konkretnih primerov spletnih iger tega tipa ter predstavili koncept s področja teorije grafov, na katerem izziv temelji. Za enostavnejše razumevanje slednjega najprej predstavljamo nekaj osnovnih pojmov iz teorije grafov.

2 OSNOVNI POJMI S PODROČJA TEORIJE GRAFOV

Teorija grafov se ukvarja s preučevanjem struktur, ki jih imenujemo grafi.

Graf je struktura, ki jo sestavljata neprazna množica elementov, imenovanih **vozlišča** ali **točke** grafa, ter množica (nekaterih) neurejenih parov teh elementov, ki jim pravimo **povezave** grafa. Množico vozlišč grafa G označimo z $V(G)$, množico povezav pa z $E(G)$. V prispevku se bomo omejili na obravnavo enostavnih grafov (razen, če bo navedeno drugače), torej grafov, ki nimajo večkratnih povezav med pari vozlišč in tudi nimajo vozlišč, povezanih samih s seboj. Za poljubni vozlišči $u, v \in V(G)$, neurejeni par $\{u, v\}$ krajše označimo kot uv . Če je $uv \in E(G)$ pravimo, da sta vozlišči u in v **sosednji** ali **povezani**. V tem primeru je vozlišče u **sosed** vozlišča v oziroma v je sosed vozlišča u . Povezani vozlišči u in v pripadata povezavi uv in sta njeni **krajišči**. Slika 1 prikazuje primer grafa G z množico vozlišč $V(G) = \{a, b, c, d, e, f\}$ in množico povezav $E(G) = \{ab, ac, ad, bc, cd, ce, de, ef\}$. Primer dveh vozlišč, ki sta soslednji oziroma povezani, sta vozlišči c in d .

Sprehod v grafu G je zaporedje vozlišč $v_1, v_2, \dots, v_k$ grafa G z lastnostjo, da za vsak $i \in \{1, 2, \dots, k-1\}$ velja $v_i v_{i+1} \in E(G)$. V primeru, da so vsa vozlišča sprehoda paroma različna, tak sprehod imenujemo **pot** v grafu G . Če za dani sprehod velja, da začetno in končno vozlišče tega sprehoda sovpadata, potem se ta sprehod imenuje **obhod**. Za lažje razumevanje zapisanega navajamo še primere sprehoda, poti in obhoda v grafu G , ki je prikazan na sliki 1. Primer sprehoda v grafu G : a, b, c, e, d, c, a, d ; primer obhoda v grafu G : a, b, c, e, d, c, a ; primer poti v grafu G : a, b, c, e, d .

Slika 1: Graf G .

3 PRIMERI SPLETNIH IZZIVOV, KI VKLJUČUJEJO REŠEVANJE PROBLEMOV S PODROČJA TEORIJE GRAFOV

V tem razdelku bomo predstavili nekaj izbranih (spletnih) izzivov, pri razreševanju katerih rešujemo probleme s področja teorije grafov. Vsak izmed opisanih izzivov ima to lastnost, da ga lahko razrešimo četudi nimamo znanja s področja teorije grafov; pravzaprav je vsak izziv tudi predstavljen in podan na način, ki ne zajema (predvsem z vidika terminologije) konceptov s področja teorije grafov.

Izbrane probleme, naloge oziroma izzive bomo najprej opisali. Nato bomo za vsakega od njih podali tudi konkretne primere spletnih iger oziroma izzivov, ki vključujejo reševanje tega problema. Vsak problem bomo osvetlili tudi z vidika teorije grafov. Za vsako nalogo bomo namreč predstavili koncept s področja teorije grafov, na katerem ta naloga temelji, ter prikazali problem z omenjenega področja, ki je analogen problemu dane naloge.

V prispevku so sicer obravnavani spletni izzivi, a prav vsakega od teh problemov lahko razrešimo tudi na bolj klasičen način, z uporabo lista in pisala. Dodano vrednost, ki jo prinašajo spletne različice izzivov, zaznavamo predvsem v dveh aspektih.

- Motivacijski vidik. V spletnih različicah izzivov se uporabnik spopade z več nalogami istega tipa, ki so urejene glede na zahtevnost v različne nivoje oziroma stopnje. Razrešitev posameznega nivoja omogoča pridobitev nove naloge in s tem novega izziva, kar lahko uporabnika motivira za delo. Nekateri spletni izzivi lahko na motivacijo uporabnika vplivajo tudi s svojo grafično podobo in dodatnimi elementi (nagrade, točke, omejitve časa).
- Poskušanje in razvijanje strategije. Spletno okolje ponuja več enostavnejših možnosti za poskušanje razreševanja naloge, s tem pa tudi dodatne opcije za razvijanje in preizkušanje novih strategij.

Problemi, naloge, uganke oziroma izzivi, ki jih opisujemo v nadaljevanju, so naslednji: Prečkanje reke, Nariši tako, da se črte ne sekajo, Nariši sliko v eni potezi ter sudoku.

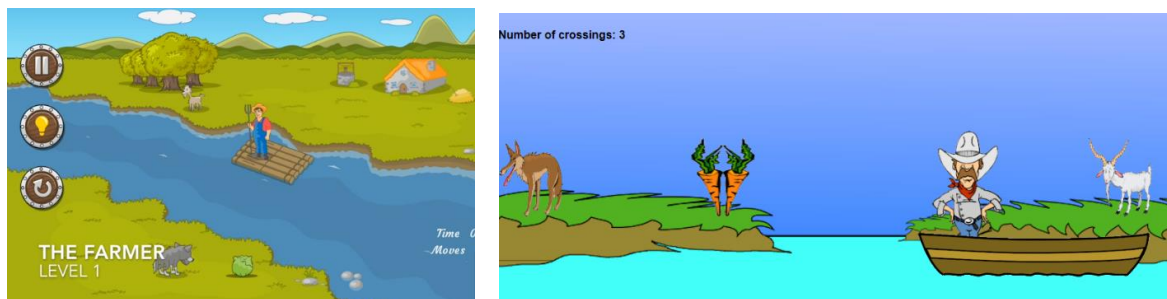
Prečkanje reke

Naloga igralca pri različnih variacijah problema prečkanja reke je ta, da opiše oziroma načrtuje potek transporta več oseb, živali ali objektov preko reke, pri čemer mora upoštevati dane pogoje. Ti se v veliko primerih nanašajo na to, koliko oseb, živali ali objektov lahko hkrati prepeljemo čez reko ter na to, katere kombinacije oseb, živali ali objektov na istem bregu reke ali v čolnu v nekaterih primerih niso dovoljene.

Spletne naloge v zvezi s prečkanjem reke običajno ponujajo nabor različnih variacij izziva prečkanja reke, ki so urejene po težavnosti in posledično razporejene na različne ravni.

Eden izmed zelo znanih problemov v zvezi s prečkanjem reke, ki nastopi v veliko spletnih izzivih, je sledeči problem, poznan tudi pod imenom *Volk, koza, zelje*. Kmet je skupaj z volkom, kozo in zeljem na istem bregu reke. Kozo, volka in zelje želi s čolnom prepeljati na drugi breg te reke. Čoln je majhen, zato je poleg kmeta lahko v njem le ena žival ali pa samo zelje, v vsakem primeru pa vožnje brez kmeta niso dovoljene. Poleg tega mora kmet transport izvesti tako, da hkrati brez nadzora kmeta ne smeta biti na istem bregu reke volk in koza (kajti volk lahko poje kozo) ter enako koza in zelje (ker lahko koza poje zelje). Naša naloga je, da načrtujemo prečkanje reke, ki zadosti vsem zapisanih zahtevam, če seveda to obstaja.

Dve izmed spletnih iger oziroma izzivov, ki med drugim ponujata razreševanje problema *Volk, koza, zelje*, sta igri *River Crossing – Logic Puzzles* [13] in *River Crossing Challenge* [14] (zaslonska posnetka obeh iger sta prikazana na sliki 2).



Slika 2: Igra *River Crossing – Logic Puzzles* [13] ter igra *River Crossing Challenge* [14].

Problem *Volk, koza, zelje* lahko rešimo s (sistematičnim) poskušanjem, z logičnim razmišljanjem in dobro premišljenim načrtovanjem prečkanja brez poznavanja teorije grafov (običajno je ta problem tudi predstavljen na takšen način).

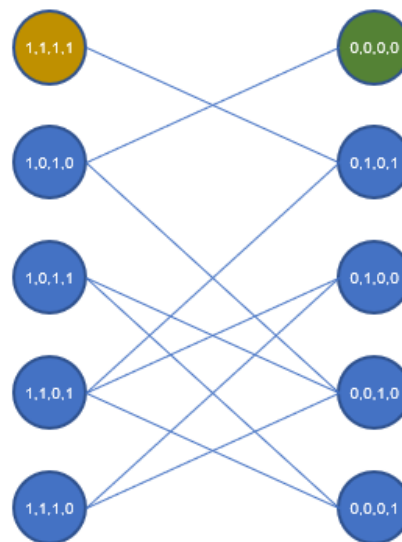
Druga možnost reševanja pa je predstavitev situacije s pomočjo grafa na sledeči način. Vozlišča grafa naj bodo vsi binarni nizi iz štirih elementov (vsak tak niz je pravzaprav zaporedje štirih števil, vsako od teh števil pa je bodisi 0 ali 1). Vsaka koordinata niza predstavlja enega izmed udeležencev v problemu (1. koordinata: kmet, 2. koordinata: volk, 3. koordinata: koza, 4. koordinata: zelje), vrednost 0 v nizu pomeni nahajanje na začetnem bregu reke, vrednost 1 pa na ciljnem bregu reke. Ker se koza in zelje ne smeta nahajati skupaj na istem bregu reke, če tam ni kmeta, nas vozlišča $(1,0,0,0)$, $(1,1,0,0)$, $(0,0,1,1)$ in $(0,1,1,1)$ ne zanimajo, prav tako na podlagi podobnega razloga v zvezi s kozo in volkom ugotovimo, da tudi vozlišča $(1,0,0,1)$ in $(0,1,1,0)$ ni treba obravnavati. Med preostalimi vozlišči grafa narišemo povezave tako, da sta poljubni dve vozlišči povezani natanko tedaj, ko opisujeta situaciji, ki sta lahko zaporedni v procesu prečkanja reke. Na ta način ustvarimo graf tako, da je rešitev naloge prečkanja reke ekvivalentna iskanju poti (oziroma sprehoda) med vozliščema $(0,0,0,0)$ in $(1,1,1,1)$ v tem

grafu (omenjeni dve vozlišči namreč predstavljata začetno ter želeno stanje). Opisani graf je predstavljen na sliki 3.

Kot smo omenili, spletni izzivi v zvezi s prečkanjem reke običajno združujejo več posameznih nalog oziroma več variacij tega izziva. V nadaljevanju navajamo dve različici.

- Preko reke želimo prepeljati tri zakonske pare (tri može in tri žene), pri čemer sta omejitvi naslednji: v čolnu sta lahko hkrati največ dve osebi ter nobena žena ne sme biti v prisotnosti moškega, ki ni njen mož, če ta ni poleg. Takšno igro ponuja recimo *The River Tests - IQ Logic Puzzles & Brain Games* [17].
- Preko reke želimo prepeljati štiri popotnike. Podane so težje omenjenih oseb, naloga igralca pa je, da načrtuje prečkanje reke vseh oseb, pri čemer ima čoln omejitev glede dovoljene težje (primer spletne igre: *River Crossing Puzzle* [15]).

Pri obeh opisanih problemih gre sicer za nalogo prečkanja reke, a problema se razlikujeta glede na dane zahteve in pogoje, ki jih moramo upoštevati pri reševanju. Pri prvi variaciji naloge imamo namreč šest oseb, ki potujejo čez reko, pri 2. pa le štiri; pri 1. nalogi ima čoln omejitev glede števila oseb, pri 2. pa glede teže teh. Različnost teh zahtev in omejitev direktno ne kaže na analogijo postopkov reševanja opisanih problemov. Če pa na problema pogledamo z vidika teorije grafov, ugotovimo, da lahko opisani situaciji v obeh problemih predstavimo z grafom na podoben način kot v primeru problema *Volk, koza, zelje*. Nalogi tako preoblikujemo v nalogo iskanja poti v grafu. Navedeno torej pokaže, da kljub različnima nalogama, ki na prvi pogled ne implicirata enakih strategij reševanja, z vidika teorije grafov obe predstavljata enak problem.



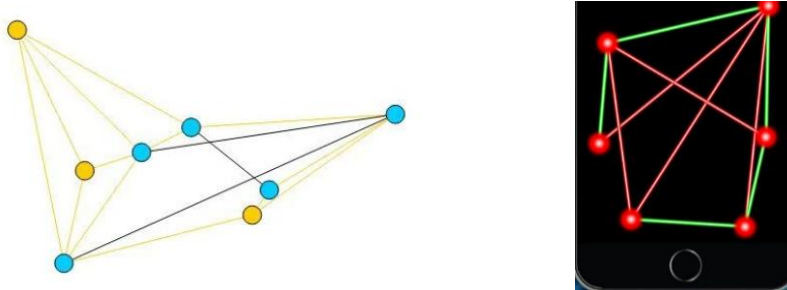
Slika 3: Graf, ki opisuje situacijo iz problema *Volk, koza, zelje* (vir slike: [1]).

Nariši tako, da se črte ne sekajo

Pri tem izzivu je podan graf (četudi ni poimenovan kot graf), ki je narisano tako, da se (nekateri) povezave tega grafa sekajo (imajo skupne točke, ki niso krajišča povezav). Naloga reševalca je,

da s premikanjem vozlišč grafa (točk v ravnini) ustvari takšno sliko grafa, kjer nobeni dve povezavi nimata skupnih točk, razen morebitnih skupnih vozlišč. Pri igrah omenjenega tipa se običajno povezave, ki nimajo presečišč z drugimi povezavami, pobarvajo drugače kot tiste povezave, ki se sekajo s katero izmed preostalih povezav.

Na spletu lahko najdemo več izzivov omenjenega tipa, na primer: *Planar Puzzles* [11] ali *Razpletati vozle: logika* [12]. Omenjeni igri oziroma izziva sta prikazana na sliki 4. V večini primerov nalog omenjenega tipa gre za podobne variacije teh; razlike lahko opazimo pri težavnosti, kar sledi iz uporabe grafov z različnimi lastnostmi.



Slika 4: Igra *Planar Puzzles* [11] in igra *Razpletati vozle: logika* [12].

Opisan izziv lahko umestimo med strateške naloge – tekom razreševanja različnih primerov (izvajanja dejavnosti z različnimi grafi) je pričakovano, da uporabnik razvije nekaj smiselnih strategij oziroma načel, ki jim pri reševanju sledi (na primer, v nekaterih primerih je smiselno, da po ravnini premikamo vozlišče, ki je krajišče največ takšnih povezav, ki določajo presečišča z drugimi povezavami).

Z znanjem s področja teorije grafov pri hitrem pogledu na opisani izziv ugotovimo, da cilj tega sovпада z idejo risanja **ravninskih risb** grafa; to so takšne risbe grafov v ravnini, kjer poljubni dve povezavi grafa nimata skupnih točk, razen morebitnega skupnega krajišča. Povedano drugače, graf narišemo v ravnini tako, da se poljubni dve povezavi ne sekata. Vseh grafov ne moremo narisati na omenjeni način; tiste, ki jih lahko, imenujemo **ravninski grafi**.

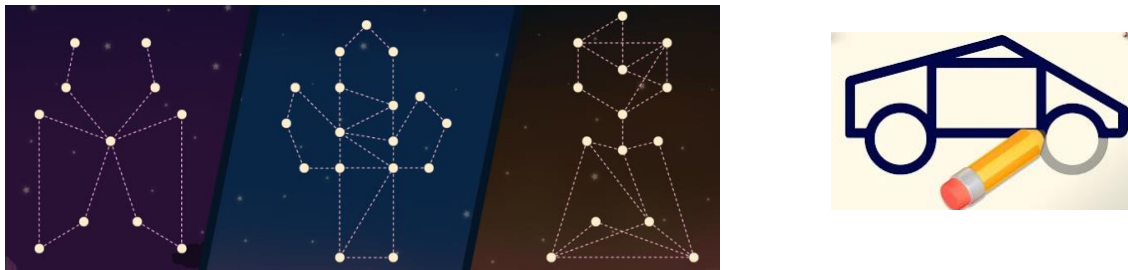
Pri omenjenem izzivu torej obravnavamo slike, ki predstavljajo ravninske grafe, četudi to ni razvidno na prvi pogled. Naš cilj je narisati ravninsko risbo danega grafa.

Nariši sliko v eni potezi

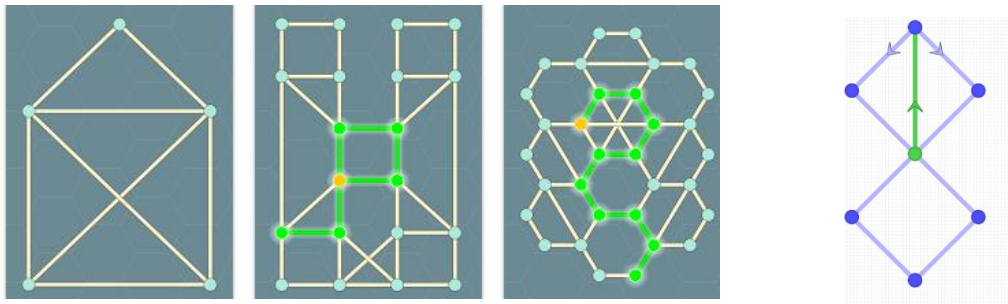
Kot že ime izziva pove, je glavni cilj tega narisati podano sliko, skico, krivuljo, lomljeno črto, ... oz. skupek teh le v eni potezi. To pomeni, da z risanjem pričnemo v poljubno izbrani točki, nato pa s pisalom sledimo vnaprej narisanim oziroma predvidenim črtam, pri čemer moramo upoštevati naslednja pravila:

- s pisalom rišemo neprekinjeno črto (od začetka do konca risanja tega ne prekinemo);
- ko narišemo črto, ponovno risanje iste črte ali dela že narisane črte ni dovoljeno;
- narisati moramo vse predvidene črte;
- nekatere izmed omenjenih iger zahtevajo, da risanje zaključimo v točki, kjer smo pričeli.

Na voljo je zelo veliko spletnih iger, ki zahtevajo risanje slike v eni potezi (na primer: *One Touch Drawing - One Line* [10], *One Line Drawing - Brain Test* [7], *One Touch Draw* [8], *One touch Drawing* [9]), zaslonske slike nekaterih prikazujemo na slikah 5 in 6.



Slika 5: *One Touch Drawing - One Line* [10] in *One Line Drawing - Brain Test* [7].



Slika 6: *One Touch Draw* [8] in *One touch Drawing* [9].

Igre, ki zahtevajo, da narišemo sliko v eni sami potezi, povezuje skupen cilj, matematično ozadje in osnovno navodilo igre, vseeno pa obstaja več variacij takšnih iger. V nadaljevanju omenjamo nekaj od teh.

- Kot smo že omenili zgoraj, nekatere izmed iger zahtevajo, da risanje zaključimo tam, kjer smo z risanjem pričeli. Nekatere igre te zahteve ne navajajo.
- Pri nekaterih igrah je podana smer dela poti, ki jo moramo pri risanju upoštevati. V nekaterih primerih ta določena smer ni konstantna ali vpliva na zahtevane smeri risanja drugih delov poti. Primer takšne variacije nastopi pri igri *One touch Drawing* [9].

Risanje slike v eni potezi poteka v več korakih: na vsakem od teh moramo dobro premisliti, ali je naslednji korak smiseln in omogoča nadaljevanje našega dela – hitro se namreč zgodi, da na nekem koraku risanja ne moremo nadaljevati, četudi še nismo narisali celotne slike. Razreševanje izziva tako zahteva logično in strateško razmišljanje ter sprotno načrtovanje dela.

Naloga, da narišemo sliko v eni potezi, je tesno povezana s konceptom Eulerjevih oziroma Poleulerjevih grafov. Omenjena koncepta in sorodne predstavljamo v nadaljevanju.

Naj bo G poljuben graf. Sprehod v grafu G imenujemo **Eulerjev sprehod**, če v tem sprehodu vsaka povezava grafa G nastopi natanko enkrat. Eulerjev sprehod torej poteka preko vseh povezav grafa, a nobeni dve povezavi tega sprehoda nista enaki (nobena povezava se ne pojavi dvakrat ali večkrat). Če v Eulerjevem sprehodu začetno in končno vozlišče sovpadata, je ta sprehod pravzaprav obhod. V tem primeru ga imenujemo **Eulerjev obhod**. Če graf G premore Eulerjev obhod, rečemo, da je ta graf **Eulerjev**. Če graf G ne premore Eulerjevega obhoda, premore pa Eulerjev sprehod, potem ta graf imenujemo **Poleulerjev graf**.

Če želimo razrešiti zgoraj predstavljen izziv in dano sliko narisati v eni potezi ter zaključiti risanje v točki, kjer smo pričeli, moramo pravzaprav narisati Eulerjev obhod tistega grafa, ki ustreza podani risbi. Vozlišča tega grafa so običajno narisana kot točke (pike, krožci) na sliki, če pa niso, so to krajišča daljic ali krivih črt na sliki oziroma presečišča (stičišča, dotikališča) teh. Povezave grafa ustrezajo črtam na sliki.

Če pri risanju slike z eno potezo ni zahtevano, da začetek in zaključek risanja sovpadata, potem rišemo Eulerjev sprehod grafa, ko razrešujemo podani izziv.

Sudoku

Sudoku je ena izmed zelo znanih miselnih ugank. Uganka (klasični sudoku) je predstavljena na podlagi, ki sestoji iz kvadrata dimenzije $n \cdot n$, sestavljenega iz n^2 enotskih kvadratkov oziroma polj, v nekaterih izmed teh pa so vnaprej zapisana števila. Dodatno so polja na podlagi združena v kvadrate – vsak izmed teh združuje n polj. Cilj igralca je zapolniti vsa prazna polja na podlagi s števili iz množice $\{1, 2, \dots, n\}$, pri čemer je pomembno, da vsako število nastopi v vsaki vrstici, v vsakem stolpcu ter v vsakem kvadratu natanko enkrat.

Slika 7 (levo) prikazuje sudoku dimenzije $9 \cdot 9$ – podlaga je sestavljena iz 81 polj, ki so razporejena v 9 vrstic, 9 stolpcev oziroma 9 kvadratov. Kot je razvidno s slike, moramo na primer pri vpisovanju številke v modri kvadrat upoštevati to, da se številka, ki jo bomo zapisali, ne pojavi v nobenem od sivo označenih polj.

Opisano idejo uganke sudoku lahko variiramo na različne načine: števila lahko pripadajo različnim različno močnim številskim množicam (lahko rešujemo sudoku z 9 števili, kot kaže primer, ali manj oz. več), števila lahko nadomestimo z drugačnimi elementi (na primer z barvami ali z liki), obliko kvadrata lahko nadomestimo z drugimi oblikami in podobno.

Na spletu najdemo zelo veliko različic klasične igre sudoku (dimenzije podlage $9 \cdot 9$, kjer v polja vpisujemo številke) ter tudi veliko različnih drugačnih variacij te igre. Slika 7 (desno) prikazuje eno variacijo omenjene spletne naloge.



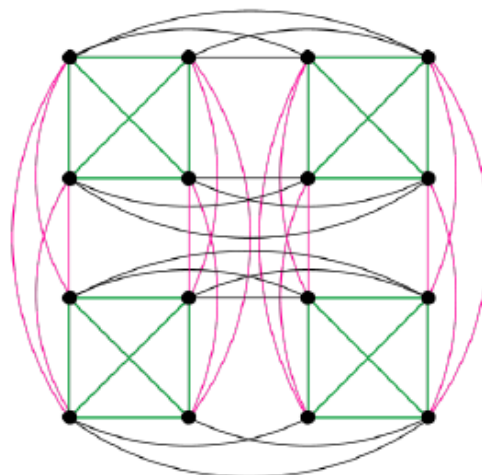
Slika 7: Igra sudoku Game – Daily Puzzles [16] in igra Color Sudoku - Sudoku Puzzles [3].

Klasično uganke sudoku v splošnem razrešimo tako, da preverimo, katera že vpisana števila nam natanko določajo tista števila, ki v polja še niso vnesena. Da lahko to izvajamo, moramo imeti natančen in tudi sistematičen vpogled v trenutno stanje že vpisanih števil. Kadar omenjena strategija igre ne zadošča za reševanje, se lahko na primer poslužujemo tudi izločanja števil, ki na neko polje gotovo ne sodijo, ali pa na kakšnem koraku izvedemo tudi poskušanje.

Dano uganke lahko razrešimo tudi s pomočjo teorije grafov. Prvi pogled na igralno podlago uganke sudoku morda ne razkriva ozadja te s področja teorije grafov, a podlago lahko zlahka predstavimo kot graf, sudoku pa razrešimo z uporabo konceptov s področja teorije grafov.

Podlago sudoku predstavimo z grafom na naslednji način: vozlišča grafa ustrezajo poljem na podlagi, dve vozlišči pa povežemo natanko tedaj, ko sta pripadajoči polji sudokuja v isti vrstici, istem stolpcu ali istem kvadratu. K tistim vozliščem, ki ustrezajo poljem, v katerih so že vpisana števila, ta števila pripišemo. Slika 8 prikazuje graf, ki ustreza igralni podlagi sudokuja, ki sestoji iz 16 polj.

Reševanje sudokuja je ekvivalentno dobremu barvanju vozlišč pripadajočega grafa, v katerem so nekatera vozlišča že pobarvana (imajo že dodeljeno številko oziroma barvo). **Dobro barvanje vozlišč grafa** G je namreč definiran kot preslikava, ki vsakemu vozlišču tega grafa priredi eno barvo (te običajno označujemo z naravnimi števili) na način, da poljubni dve sosednji vozlišči grafa prejmeta različni barvi (oziroma števili). Reševanje sudokuja s podlago dimenzije $n \cdot n$ je tako ekvivalentno dobremu barvanju vozlišč pripadajočega grafa z n barvami. Ker so v grafu vozlišča iste vrstice (stolpca oziroma kvadrata) povezana, bomo v posamezni vrstici (stolpcu oziroma kvadratu) podlage sudokuja imeli navedeno posamezno številko največ enkrat (oziroma natanko enkrat, saj je polj v posamezni vrstici, stolpcu oz. kvadratu ravno toliko, kot imamo na voljo barv).



Slika 8: Graf, ki ustreza podlagi miselne uganke *sudoku* (dimenzije $4 \cdot 4$) [4].

4. ZAKLJUČEK

V prispevku smo predstavili spletne izzive (igre, uganke, probleme in druge naloge), pri katerih gre pravzaprav za reševanje nalog oziroma problemov s področja teorije grafov. Z razreševanjem posameznega izziva tako uporabnik posredno obravnava različne koncepte s

področja teorije grafov, ki pa niso predstavljeni v jeziku te matematične discipline. Za soočanje z omenjenimi izzivi igralci sploh ne potrebujejo znanja o grafih in povezanih konceptih, niti ni nujno, da poznajo to vejo matematike.

S prispevkom smo osvetlili uporabnost teorije grafov v spletnih igrah in posledično tudi v naših vsakdanjkih. Kot že omenjeno, izsledke te veje matematike s pridom izkoriščamo tudi na številnih različnih drugih področjih, kar kaže na njeno veliko pomembnost in prisotnost ne le v matematiki, temveč tudi v vsakdanjem življenju.

Literatura

- [1] *Bipartite Graph to solve the wolf river crossing problem*. Pridobljeno 15. 6. 2024.
<https://cs.stackexchange.com/questions/138455/bipartite-graph-to-solve-the-wolf-river-crossing-problem>
- [2] Bondy, J. A., Murty, U. S. R. (1976). *Graph theory with applications* (Vol. 290). London: Macmillan.
- [3] *Color Sudoku - Sudoku Puzzles*. Pridobljeno 27. 6. 2024.
<https://play.google.com/store/apps/details?id=com.wormlab.colorsudoku&hl=en>
- [4] Ferme, J. (2016). *Obravnava barvanj grafov in tetivnih grafov v srednješolskem izobraževanju: magistrsko delo*. Univerza v Mariboru, Fakulteta za naravoslovje in matematiko.
- [5] Ferme, J., Štesl Mesarič, D. (2023). *Teorija grafov v osnovnošolskem in srednješolskem izobraževanju*. Dianoia, 7, 17–25.
- [6] Mondal, B., De, K. (2017). *An overview applications of graph theory in real field*. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2(5), 751–759.
- [7] *One Line Drawing - Brain Test*. Pridobljeno 26. 6. 2024.
https://play.google.com/store/apps/details?id=com.fc.p.fr.brain.puzzle.oneline.draw&hl=en_US
- [8] *One Touch Draw*. Pridobljeno 26. 6. 2024.
<https://play.google.com/store/apps/details?id=com.alcamasoft.onetouchdraw&hl=en>
- [9] *One touch Drawing*. Pridobljeno 26. 6. 2024.
https://play.google.com/store/apps/details?id=com.ecapycsw.onetouchdrawing&hl=en_US
- [10] *One Touch Drawing - One Line*. Pridobljeno 26. 6. 2024.
https://play.google.com/store/apps/details?id=com.pm4.onetouch&hl=en_US.
- [11] *Planar Puzzles*. Pridobljeno 20. 6. 2024.
<https://play.google.com/store/apps/details?id=com.andromo.dev64620.app414813&hl=en>

[12] *Razpletati vozle: logika*. Pridobljeno 20. 6. 2024.

<https://play.google.com/store/apps/details?id=air.App2Eleven.LogicGamePuzzlesUntangle>

[13] *River Crossing – Logic Puzzles*. Pridobljeno 10. 6. 2024.

https://play.google.com/store/apps/details?id=com.androyal.rivercrossing.complete&hl=en_US

[14] *River Crossing Challenge*. Pridobljeno 10. 6. 2024.

https://www.transum.org/software/River_Crossing/Level1.asp

[15] *River Crossing Puzzle*. Pridobljeno 10. 6. 2024.

<https://planeta42.com/math/rivercrossing/>

[16] *Sudoku Game – Daily Puzzles*. Pridobljeno 27. 6. 2024.

<https://play.google.com/store/apps/details?id=com.gamovation.sudoku&hl=en>

[17] *The River Tests - IQ Logic Puzzles & Brain Games*. Pridobljeno 10. 6. 2024.

<https://apps.microsoft.com/detail/9nsncz4scxr?hl=sl-si&gl=SI>

Sinteza in karakterizacija kobaltove koordinacijske spojine s 4-cianopiridinom

Synthesis and characterization of cobalt coordination compound with 4-cyanopyridine ligand

Aljaž Čelofiga^a, Amalija Golobič^b, Brina Dojer^a

^aUniverza v Mariboru, Fakulteta za naravoslovje in matematiko, Koroška cesta 160, 2000 Maribor

^bUniverza v Ljubljani, Fakulteta za kemijo in kemijsko tehnologijo, Večna pot 113, 1000 Ljubljana

Povzetek

V prispevku predstavljamo sintezo in karakterizacijo nove, do sedaj še neznane kobaltove koordinacijske spojine z vezanim cianopiridinskim ligandom (4-cianopiridinom). Kobaltove koordinacijske spojine igrajo ključno vlogo kataliziranja različnih kemijskih reakcij, sodelujejo pri biokemijskih procesih ter gradijo kompleksnejše molekule organizmov. Uspešno smo sintetizirali spojino, ki ima obliko monoklinskih kristalov z oktaedrično koordiniranim kobaltovim centralnim ionom in molekulska formulo $[\text{Co}(\text{C}_2\text{H}_3\text{O}_2)_2(\text{C}_6\text{H}_4\text{N}_2)(\text{H}_2\text{O})_3]$. Karakterizacija spojine je potekala s pomočjo Fourierjeve transformirajoče infrardeče spektroskopije (FTIR), termogravimetrične analize (TGA) in rentgenske analize na monokristalu (XRD).

Ključne besede: kobalt, koordinacijske spojine, cianopiridini, termična analiza, strukturna analiza na monokristalu, FTIR analiza

Abstract

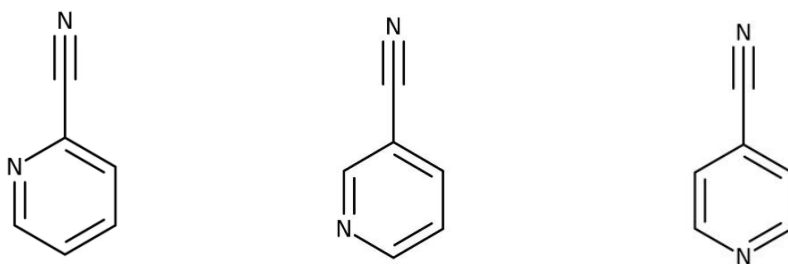
In this paper, we present the synthesis and characterization of a new, so far unknown cobalt coordination compound with a bound cyanopyridine ligand (4-cyanopyridine). Cobalt coordination compounds play a key role in catalyzing various chemical reactions, participate in biochemical processes and build more complex molecules of organisms. We have successfully synthesized a compound in the form of monoclinic crystals with octahedrally coordinated cobalt central ion and the molecular formula $[\text{Co}(\text{C}_2\text{H}_3\text{O}_2)_2(\text{C}_6\text{H}_4\text{N}_2)(\text{H}_2\text{O})_3]$. The compound was characterized using Fourier transform infrared spectroscopy (FTIR), thermogravimetric analysis (TGA) and single crystal X-ray analysis (XRD).

Key words: cobalt, coordination compounds, cyanopyridines, thermal analysis, single crystal X-ray analysis, FTIR analysis

1 UVOD

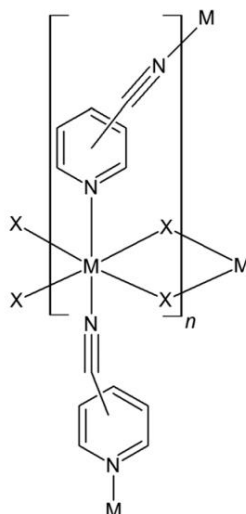
Kobaltove koordinacijske spojine so pomembne v raziskovanju katalizatorskih karakteristik v številnih biokemijskih reakcijah [1,2,3]. Kobalt kot element je v telesu sestavni del mnogih encimov, ki sodelujejo pri različnih biokemijskih procesih in je najpogosteje vezan v strukturo, značilno za koordinacijske spojine [3].

Za sintezo novih kobaltovih koordinacijskih spojin smo kot vir kobaltovih ionov uporabili kobaltov(II) acetat tetrahidrat, za ligande pa 2-, 3- in 4-cianopiridin. Slika 1 prikazuje skeletne formule izbranih cianopiridinskih ligandov.



Slika 1: Od leve proti desni: skeletne formule 2-, 3- in 4-cianopiridina.

Koordinacijske spojine 4-cianopiridina s prehodnimi kovinami (rutenijem, železom, bakrom, srebrom, zlatom, kobaltom) so znane že desetletja [4-7]. Ker vsebujejo cianopiridinski ligandi dva dušikova atoma (dve vezavni mesti), se lahko v koordinacijskih spojinah obnašajo tudi kot mostovni ligandi, s čimer se povečuje možnost tvorbe večjernih in polimernih kompleksov [8] (slika 2).



Slika 2: Prikaz možnosti tvorbe polimernih kompleksov s cianopiridinom, ki ima dve vezavni mesti.

2 EKSPERIMENTALNI DEL

Sinteze smo izvajali na tri različne načine, in sicer z mešanjem pri sobni temperaturi, z mešanjem pri povišani temperaturi in z mešanjem pod refluxom.

Sinteza, pri kateri smo dobili spodaj prikazan produkt (slika 3), je potekala z mešanjem pri sobni temperaturi. Zanj smo potrebovali kobaltov(II) acetat tetrahidrat (0,24909 g; 1 mmol) in 4-cianopiridin (0,20822 g; 2 mmol). Vsakega smo raztopili v 5 mL metanola. Raztopino liganda smo postopoma dodajali k raztopini kobaltovega(II) acetata tetrahidrata.

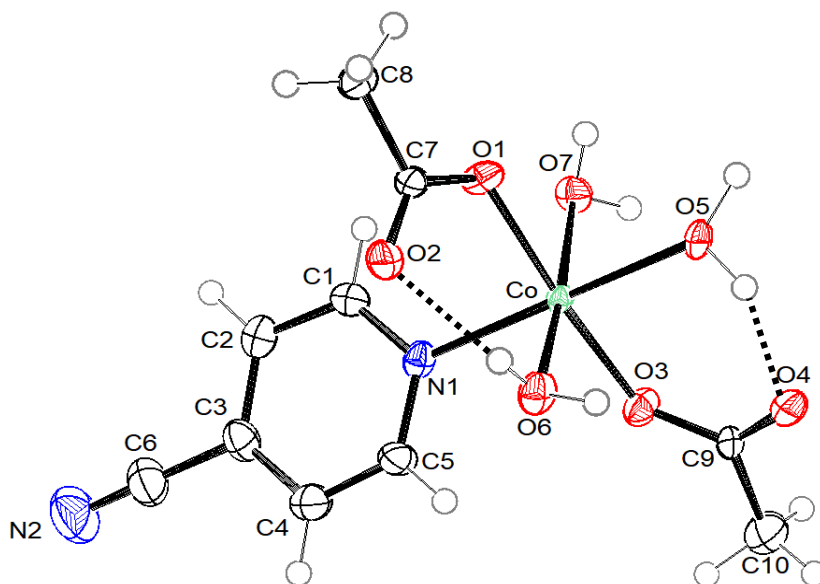
Mešanje je potekalo na magnetnem mešalu približno 15 min. Čašo s produktom sinteze smo pustili v digestoriju, dokler se niso iz raztopine izločili kristali (slika 3).



Slika 3: Izgled metanolne raztopine kobaltovega(II) acetata tetrahidrata in liganda 4-cianopiridina po mešanju pri sobni temperaturi.

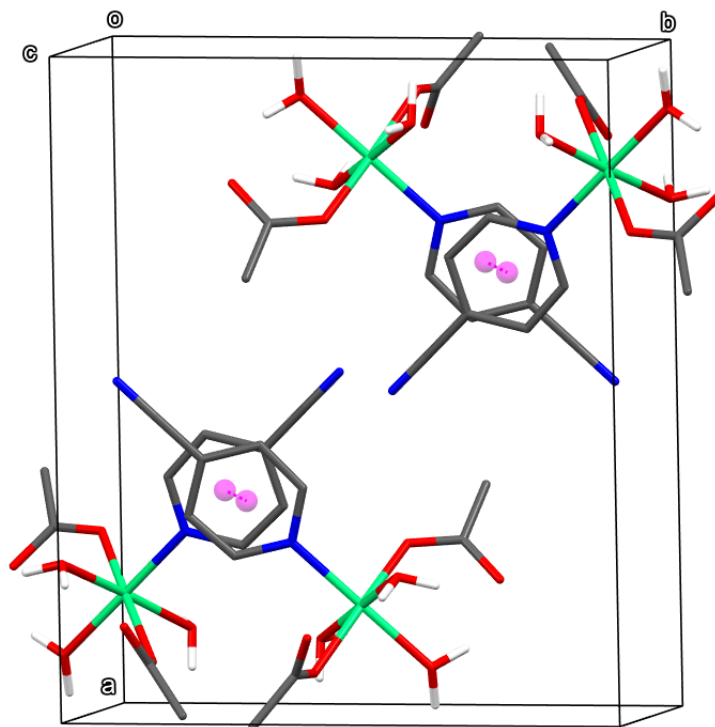
3 REZULTATI IN DISKUSIJA

Novosintetizirana koordinacijska spojina ima molekulsko formulo $C_{10}H_{16}CoN_2O_7$ in kristalizira v obliki temno rdečih kristalov. Kobaltov centralni ion je koordiniran oktaedrično, nanj je vezanih šest ligandov (ena molekula 4-cianopiridina, dve acetatni molekuli in tri molekule vode) (slika 4).



Slika 4: Struktura novosintetizirane spojine $[Co(C_2H_3O_2)_2(C_6H_4N_2)(H_2O)_3]$ s ponazorjenima intermolekularnima vodikovima vezema.

Osnovna celica je monoklinskega tipa in vsebuje 4 takšne molekule (slika 5).



Slika 5: Osnovna monoklinska celica s štirimi molekulami $[\text{Co}(\text{C}_2\text{H}_3\text{O}_2)_2(\text{C}_6\text{H}_4\text{N}_2)(\text{H}_2\text{O})_3]$. Roza krogi ponazarjajo centre heteroaromatskih obročev.

Med približno vzporednimi obroči pride do $\pi \cdots \pi$ interakcij. Razdalja med centri obročev je 3.853 Å.

Tabela 1: Kristalografski podatki spojine.

Molekulska formula novosintetizirane koordinacijske spojine	$\text{C}_{10}\text{H}_{16}\text{CoN}_2\text{O}_7$
Molska masa	335,18 g/mol
Prostorska skupina	$P2_1/c$ (14)
Kristalni sistem	monoklinski
Barva kristalov	temno rdeča
Mrežne konstante	$a = 15.1276$ (11) Å $b = 12.8133$ (8) Å $c = 7.3082$ (5) Å $\alpha = 90^\circ$ $\beta = 93.257$ (6) $^\circ$ $\gamma = 90^\circ$

Volumen osnovne celice [$\text{\AA}^3$]	1414,29 (17)
Število formulskih enot v osnovni celici	4

Osnovne informacije, pridobljene s pomočjo kristalografskih meritev, smo dobili s sodelovanjem s Fakulteto za kemijo in kemijsko tehnologijo Univerze v Ljubljani. Naprava (rentgenski difraktometer) za potek meritve izkorišča vir rentgenskih žarkov. V difraktometer je vstavljen kristal, ki se nenehno vrti. Zaradi podobnosti med valovno dolžino rentgenske svetlobe in razdaljo med posameznimi atomi v kristalni strukturi lahko na podlagi tega določimo dolžino vezi med posameznimi atomi v dani kristalni strukturi [9].

Spodaj so navedene še okvirne dolžine vezi in koti.

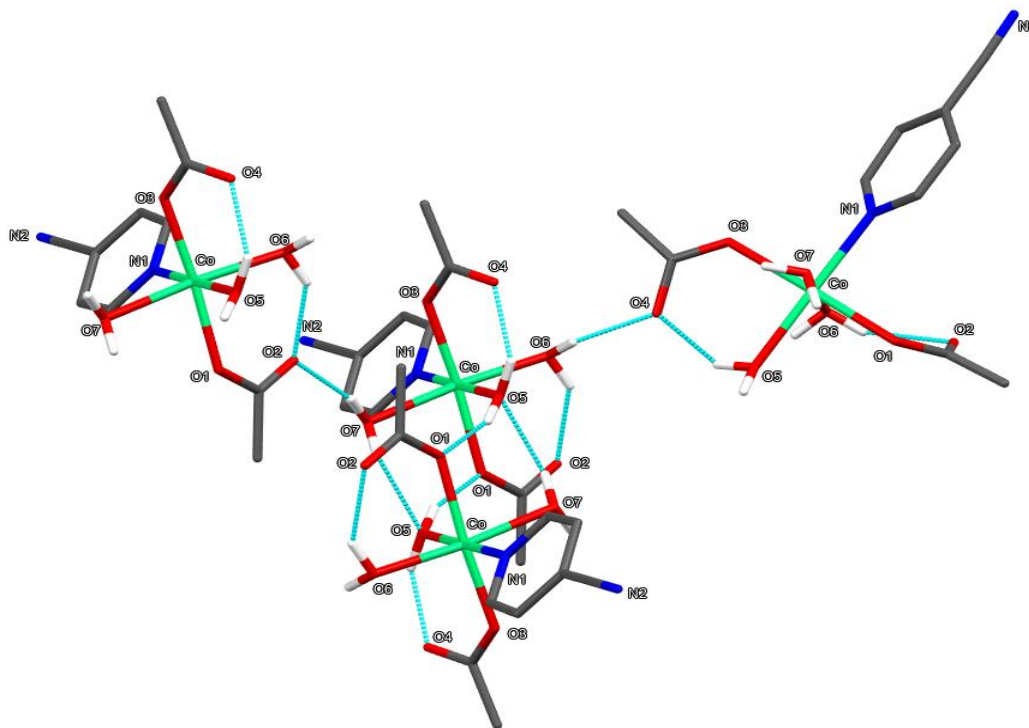
Tabela 2: Dolžine vezi v spojini.

Vez	Dolžina vezi [$\text{\AA}$]
N1 – Co	2.144 (9)
O1 – Co	2.094 (7)
O3 – Co	2.046 (8)
O5 – Co	2.122 (8)
O6 – Co	2.092 (8)
O7 – Co	2.122 (8)

Tabela 3: Koti v spojini.

Kot	Velikost kota [$^\circ$]
O3—Co—O6	93.2 (3)
O3—Co—O1	174.6 (3)
O6—Co—O1	91.9 (3)
O3—Co—O7	91.8 (3)
O6—Co—O7	175.0 (3)
O1—Co—O7	83.2 (3)
O3—Co—O5	90.6 (3)
O6—Co—O5	91.6 (3)
O1—Co—O5	87.3 (3)
O7—Co—O5	88.8 (3)
O3—Co—N1	89.7 (3)
O6—Co—N1	87.5 (3)
O1—Co—N1	92.5 (3)
O7—Co—N1	92.0 (3)
O5—Co—N1	179.1 (3)

V strukturi so tudi vodikove vezi (slika 6). Vsaka molekula vode je donor dveh O–H···O vodikovih vezi. Molekula vode z oznako O5 je z intramolekularno vodikovo vezjo povezana z O4 atomom iz acetatnega liganda istega kompleksa. Z intermolekularno vodikovo vezjo pa je povezana z O1 atomom acetatnega liganda sosednjega kompleksa. Podobno je tudi molekula vode z oznako O6 z intramolekularno vodikovo vezjo povezana z O2 atomom iz acetatnega liganda istega kompleksa ter z intermolekularno vodikovo vezjo z O4 atomom acetatnega liganda sosednjega kompleksa. Molekula vode z oznako O7 je povezana z eno intermolekularno vodikovo vezjo z molekulo vode z oznako O5 sosednjega kompleksa ter z O2 atomom acetatnega liganda sosednjega kompleksa.



Slika 6: Prikaz povezovanja koordinacijskih molekul z vodikovimi vezmi.

Tabela 4: Intra- in intermolekularne vodikove vezi v spojini.

D – H···A	D—H [Å]	H···A [Å]	D···A [Å]	D—H···A [°]
O5 – H···O4	0.87	1.84	2.599(11)	145
O5 – H···O1 ⁱ	0.99	1.76	2.715(11)	161
O6 – H···O4 ⁱⁱ	0.87	1.92	2.715(11)	151
O6 – H···O2	0.87	2.00	2.744(11)	142
O7 – H···O2 ⁱⁱⁱ	0.87	1.83	2.674(11)	164
O7 – H···O5 ⁱ	0.87	2.06	2.899(11)	161

Simetrijske kode: *i* = -*x*,1-*y*,1-*z*; *ii* = *x*,1/2-*y*,-1/2+*z*; *iii* = *x*,*y*,1+*z*

Ob merjenju in določevanju strukture se je izkazalo, da so bili kristali slabše kakovosti in posledično tudi podatki slabši.

Za karakterizacijo smo izbrali na videz primerne kristale in jih analizirali s pomočjo FTIR spektroskopije (določitev nihanj posameznih funkcionalnih skupin) in termogravimetrične analize (TGA) (izguba mase vzorca ob povišanju temperature).

S pomočjo Fourierjeve transformacije smo določili značilna nihanja vezi reprezentativnih funkcionalnih skupin v dani molekuli, saj smo jih identificirali s pomočjo značilnih nihanj, ki so pogojena z valovnimi števili [10,11].



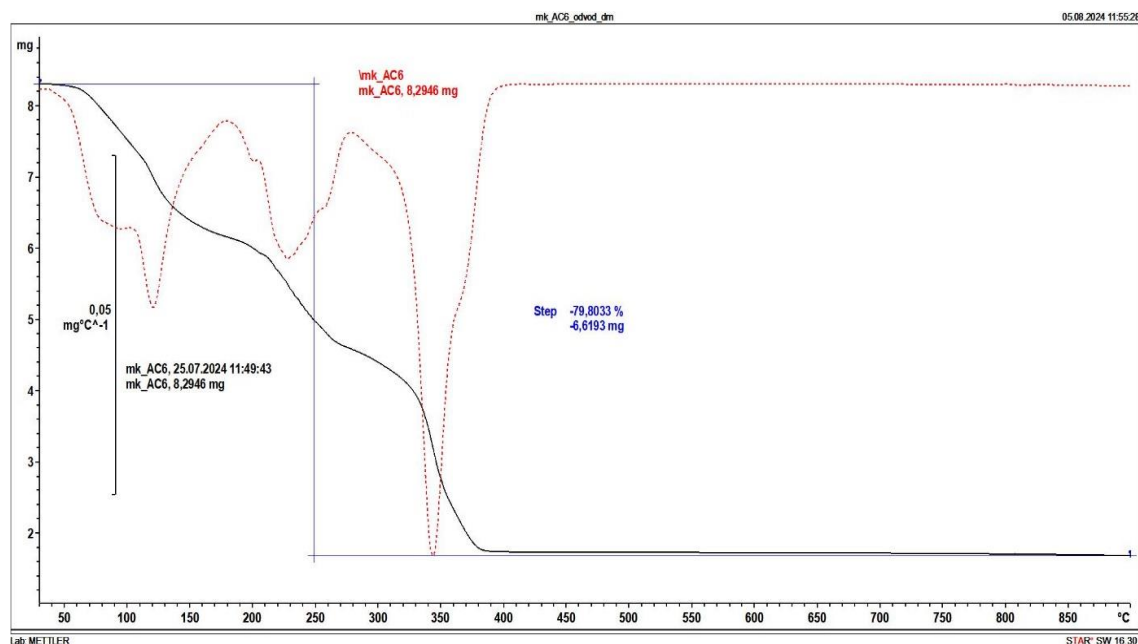
Slika 7: FTIR spekter spojine $[\text{Co}(\text{C}_2\text{H}_3\text{O}_2)_2(\text{C}_6\text{H}_4\text{N}_2)(\text{H}_2\text{O})_3]$

V območju valovnih števil okoli 3500 cm^{-1} zaznamo nihanja – OH skupin, v območju okoli $3000\text{--}3100\text{ cm}^{-1}$ (3104 , 3084 in 3029 cm^{-1}) pa nihanja – CH skupin. V območju med 2300 in 2200 cm^{-1} je nekaj vrhov, med njimi eden izrazitejši (2243 cm^{-1}), ki nakazuje nihanje $\text{C}\equiv\text{N}$ vezi. Nihanje vezi okoli 1750 cm^{-1} je pogojeno s prisotnostjo $\text{C}=\text{O}$ funkcionalne skupine. Nihanja med 1600 cm^{-1} in 1350 cm^{-1} so posledica nihanj karboksilatnih skupin. Ločimo asimetrična in simetrična nihanja omenjenih. Razlika med njimi naj bi bila pri monodentatnih acetatnih ligandih večja od $160\text{--}170\text{ cm}^{-1}$ ($\Delta = \nu_{\text{as}}(\text{COO}^-) - \nu_{\text{s}}(\text{COO}^-)$) [12,13], vendar v našem primeru ne doseže tega števila ($\nu_{\text{as}}(\text{COO}^-)$: 1544 cm^{-1} , 1498 cm^{-1} ; $\nu_{\text{s}}(\text{COO}^-)$: 1410 cm^{-1} , 1350 cm^{-1} ; $\Delta = 134\text{ cm}^{-1}$, 148 cm^{-1}). V območju okoli $1350\text{--}1280\text{ cm}^{-1}$ zaznamo nihanje C – N vezi (1287 cm^{-1}). Raztezna nihanja $\text{C}\equiv\text{N}$ vezi se pojavijo med 1200 in 1000 cm^{-1} . Deformacijska nihanja *para*-substituiranih aromatskih obročev se pojavljajo v območju med 850 in 790 cm^{-1} (830 cm^{-1}).

Termogravimetrična analiza (TGA) je analizna metoda, pri kateri proučujemo izgubo mase vzorca v odvisnosti od temperature. Med procesom segrevanja se neprekinjeno določa masa vzorca (nenehno tehtanje), ko se preko njega spusti tok inertnega plina (atmosfera ni nujno vedno inertna). Naprava je sestavljena iz peči, ki dosega visoke temperature, in tehtnice, ki se

nahajata v zaprtem sistemu. Analizo smo izvedli na Fakulteti za kemijo in kemijsko tehnologijo Univerze v Mariboru.

Dobljeni podatki so prikazani v obliki termogravimetrične krivulje, ki predstavlja izgubo mase vzorca v odvisnosti od temperature. Za natančnejšo determinacijo posameznih točk na krivulji, pri katerih pride do izgube mase vzorca, je na grafu izrisan še odvod funkcije. Na krivulji opazimo večstopenjsko izgubo mase, ki je lahko posledica razgradnje vzorca. Začetni padec mase je lahko posledica sušenja vzorca in ni posledica kemijske reakcije [14,15].



Slika 8: Termični razpad vzorca podan v mg v odvisnosti od temperature.

Pri temperaturi približno 380 °C opazimo, da se razgradnja vzorca počasi končuje, izguba mase vzorca znaša 79,80 %.

Končnega produkta segrevanja sicer nismo analizirali, vendar sam izračun kaže, da bi lahko bil mešanica CoO in Co.

4 ZAKLJUČEK

Z reakcijo kobaltovega(II) acetata tetrahidrata in 4-cianopiridina smo dobili novo koordinacijsko spojino s kemijsko formulo $[\text{Co}(\text{C}_2\text{H}_3\text{O}_2)_2(\text{C}_6\text{H}_4\text{N}_2)(\text{H}_2\text{O})_3]$. Karakterizirali smo jo s pomočjo rentgenske monokristalne analize, IR spektroskopije in termične analize. Sinteza je bila ponovljiva, predvidevamo pa, da bi lahko s pomočjo spreminjanja še katerega od pogojev sintetizirali večjedrno spojino, katere mostovni ligand bi lahko predstavljali ali 4-cianopiridin ali acetatni ligandi.

Molekule so povezane z intra- in intermolekularnimi vodikovimi vezmi, med njimi pa delujejo tudi $\pi \cdots \pi$ interakcije.

Sinteza nove koordinacijske spojine je zanimiva za raziskovanje predvsem kot potencialne katalizatorske spojine, za pospešitev kemijskih reakcij ter testiranje njihove termične obstojnosti in biokompatibilnosti.

Literatura

- [1] Kettle, S. F. A. (1996). Physical inorganic chemistry: a coordination chemistry approach. Springer Berlin, Heidelberg.
- [2] Dojer, B. (2012). Primerjava strukturnih in magnetnih lastnosti kobaltovih(II) in nikljevih(II) acetatov z izbranimi N – donorskimi ligandi: Doktorska disertacija, Univerza v Mariboru.
- [3] Constable, C. E. (2019). What's in a Name?—A Short History of Coordination Chemistry from Then to Now. *Chemistry*, 1 (1), 126–163.
- [4] Katz, N.E., Creutz, C. Sutin, N. (1988). 4-Cyanopyridine-bridged binuclear and trinuclear complexes of ruthenium and iron. *Inorg. Chem.*, 27, 1687–1694.
- [5] Almaraz, A.E., Gentil, L.A., Baraldo, L.M., Olabe, J.A. (1996). Mixed-Valence Cyanopyridine-Bridged Complexes of Pentacyanoferrate and Pentaammineruthenium: Electronic Structure, Stability, and Redox Reactivity. *Inorg. Chem.*, 35 (26), 7718–7727.
- [6] Lin P. et al. (2004). New 1- and 2-Dimensional Polymeric Structures of Cyanopyridine Complexes of AgI and Cu. *Inorg. Chem.*, 43, 181–188.
- [7] Baruah, S., Sarmah, N. and Das, B. K. (2020). Synthesis and Crystal Structure of [tetrakis (-caprylate)-bis-(4-cyanopyridine) copper(II)]. *International Journal on Emerging Technologies*, 11 (5) 193–196.
- [8] Heine, M., Fink, L., Schmidt, M.U. (2018). 3-Cyanopyridine as a bridging and terminal ligand in coordination polymer. *CrystEngComm*, 20, 7556–7566.
- [9] Bertin, E. P. (2013). Introduction to X-ray spectrometric analysis. Springer Science & Business Media, New York.
- [10] Berthomieu, C., Heinerwadel, R. (2009). Fourier transform infrared spectroscopy. *Photosynth Res*, 101, 157–170.
- [11] Bellamy, L. J. (1975). The Infra-red Spectra of Complex Molecules, Chapman and Hall, London.
- [12] Vargová, Z., Zeleňák, V., Čisárová, I., Györyová K. (2004). Correlation of thermal and spectral properties of zinc (II) complexes of pyridinecarboxylic acids with their crystal structures. *Thermochim. Acta* 423 (1–2), 149–157.
- [13] Martini, D., et al. (2002). Synthesis, spectroscopic and structural characterization of Cu(II) derivatives of tris(pyrazol-1-yl)methanes. *Inorg. Chim. Acta*, 333, 72–82.
- [14] Lothenbach, B., Durdzinski, P., & De Weerd, K. (2016). Thermogravimetric analysis. A practical guide to microstructural analysis of cementitious materials, 1, 177–211, Taylor&Francis group, London.
- [15] Wanjun, T, Donghua, C. (2007). Mechanism of Thermal Decomposition of Cobalt Acetate Tetrahydrate. *Chem. Pap.*, 61 (4) 329–332.

Tehnična analiza delnic: indikatorji in mere tveganja v programskem jeziku Python

Technical Analysis of Stocks: Indicators and Risk Measures in Python

Maša Galun¹, Mia Molnar¹, Ema Smolič¹,
Timotej Jagrič², Dušan Fister³

¹ Univerza v Mariboru, Fakulteta za naravoslovje in matematiko, Koroška cesta 160, 2000 Maribor

² Univerza v Mariboru, Ekonomsko-poslovna fakulteta, Razlagova 14, 2000 Maribor

³ Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor

Povzetek

V članku smo raziskali ključne tehnične indikatorje in mere tveganja, ki jih vlagatelji uporabljajo pri sprejemanju naložbenih odločitev. Poudarili smo vlogo indikatorjev SMA, EMA, RSI, MFI, MACD, Aroon in Hurstov eksponent pri prepoznavanju trendov in napovedovanju obratov na trgih. Poleg tega smo spoznali tudi mere tveganja beta, Sharpeov in Traynorjev količnik, ki vlagateljem omogočajo oceno tveganja v primerjavi s pričakovanim donosom. Članek ponuja vpogled v uporabo teh orodij in prikazuje, kako lahko pomagajo pri optimizaciji donosnosti ter obvladovanju tveganj v naložbah. Z vključitvijo praktičnih nalog smo bralcem omogočili, da se seznanijo z osnovami teh analiz, kar lahko pripomore k učinkovitejšemu upravljanju naložbenih portfeljev.

Ključne besede: delnice, trgovanje, indikatorji, mere tveganja, Python

Abstract

In this article, we explored key technical indicators and risk measures used by investors in making investment decisions. We highlighted the role of indicators SMA, EMA, RSI, MFI, MACD, Aroon and Hurst exponent in identifying trends and predicting market reversals. Additionally, we focused on important risk measures, such as beta, Sharpe ratio, and Treynor ratio, which allow investors to assess risk relative to expected returns. The article provides an overview of the use of these tools and demonstrates how they can assist in optimizing returns and managing risks in investments. By including practical exercises, we enabled readers to familiarize themselves with the basics of these analyses, which can contribute to better management of investment portfolios.

Key words: stocks, trading, indicators, risk measures, Python

1 Uvod

V svetu financ so pravilno sprejete odločitve ključne za uspešno naložbeno strategijo. Ena od najpogostejše uporabljenih metod za analizo trga je tehnična analiza, ki preučuje zgodovinska cenovna gibanja in druge tržne podatke. V tem članku bomo podrobneje spoznali najpogostejše uporabljene tehnične indikatorje in njihovo delovanje. Osredotočili se bomo na indikatorje SMA, EMA, RSI, MFI, MACD, Aroon in Hurstov eksponent, ki vlagateljem pomagajo prepoznati trende in morebitne obrate na trgih. Poleg tega bomo spoznali tudi mere tveganja beta, Sharpeov in Traynorjev količnik, ki omogočajo oceno tveganja naložbe

E-mail naslovi: galunmasa@gmail.com (Maša Galun¹), miamolnar02@gmail.com (Mia Molnar¹), ema.smolic@gmail.com (Ema Smolič¹), timotej.jagric@um.si (Timotej Jagrič²), dusan.fister@um.si (Dušan Fister³)

v primerjavi z njenim pričakovanim donosom. Skupaj bomo preučili, kako se ti indikatorji in mere tveganja uporabljajo pri analizi naložb, ter skozi praktične naloge pridobili boljše razumevanje uporabe teh orodij pri sprejemanju informiranih in premišljenih naložbenih odločitev.

2 Indikatorji tehnične analize

Pri analiziranju finančnih podatkov se pogosto uporabljajo indikatorji, ki finančnim analitikom in vlagateljem pomagajo sprejemati odločitve o trgovanju. Indikator je orodje, ki s pomočjo zgodovinskih podatkov o cenah in količini trgovanja napoveduje gibanja na finančnem trgu. S pomočjo indikatorjev lahko hitreje zaznamo trende, ki predstavljajo gibanje cen delnic v določenem časovnem obdobju. To omogoča učinkovitejše spremljanje trenda na trgu in hitrejši odziv na borzne dogodke. Pri izračunu tehničnih indikatorjev se uporabljajo različne vrste cen. Najpogosteje uporabljena je cena zaprtja oz. zaključna cena. Ta odraža končno vrednost delnice v določenem obdobju. Običajno si izračunane vrednosti indikatorja predstavimo z linijskim diagramom. [1]

Spoznavmo nekaj najpogosteje uporabljenih indikatorjev tehnične analize.

SMA (Preprosto drseče povprečje):

Indikator SMA (ang. "Simple Moving Average") izračuna povprečno ceno delnice, običajno z uporabo cen zaprtja, v določenem obdobju dni. Pri izračunu SMA si dolžino obdobja izberemo sami (najpogostejše vrednosti so 12, 26, 50 in 200 dni), seveda pa izbira števila dni vpliva na naravo trenda, ki ga prikaže indikator. Tako lahko pri izračunu indikatorja SMA na krajših obdobjih opazujemo kratkoročne trende, pri izračunu na daljših obdobjih pa opazujemo dolgoročne trende [8].

V praksi se glede na indikator SMA pogosto uporabljata dve različni trgovalni strategiji. Prva strategija pove, da delnico kupimo, ko se cena delnice dvigne nad vrednost SMA. Ko cena delnice pade pod SMA, je to znak za prodajo delnice. Naslednja strategija za trgovalne signale uporablja SMA, izračunan na kratkem obdobju (npr. 50 dni), in SMA, izračunan na dolgem obdobju (npr. 200 dni). Strategija pravi, da delnico kupimo, ko se kratkoročni SMA vzpne nad dolgoročni SMA. Podobno delnico prodamo, ko kratkoročni SMA pade pod dolgoročnega [8].

Formula za izračun indikatorja SMA:

$$SMA_t = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i},$$

kjer je:

- SMA_t : preprosto drseče povprečje na dan t .
- n : obdobje, za katero se izračunava SMA.
- P_{t-i} : cena delnice na dan $t - i$, kjer t pomeni trenutni dan, i pa število dni v preteklosti.

EMA (Eksponentno drseče povprečje):

Indikator EMA (ang. "Exponential Moving Average") pri izračunu uporablja podobno formulo kot SMA, le da vključuje faktor glajenja (ang. "smoothing factor"). To pomeni, da

večjo težo pripisuje najnovejšim cenam. Zaradi tega se EMA hitreje odziva na spremembe cen kot preprosto drseče povprečje (SMA), ki cene obravnava enakomerno. Trgovalne strategije so podobne strategijam pri uporabi indikatorja SMA. V splošnem se izračunata 50- in 200-dnevna EMA. Prva strategija pravi, da ko cena delnice preseže svoje 200-dnevno drseče povprečje, je to signal, da je prišlo do pozitivnega trenda [2].

Naslednja strategija pravi, da delnico kupimo, ko se kratkoročni EMA vzpne nad dolgoročni EMA. Podobno delnico prodamo, ko kratkoročni EMA pade pod dolgoročnega [4].

Formula za izračun indikatorja EMA:

$$EMA_{1,n} = SMA_1,$$

$$EMA_{t,n} = (P_t \times \alpha(n)) + (EMA_{t-1,n} \times (1 - \alpha(n))),$$

kjer je:

- $EMA_{t,n}$: eksponentno drseče povprečje na dan t .
- P_t : cena delnice na dan t .
- $\alpha(n)$: eksponentni množitelj, ki se izračuna kot:

$$\alpha(n) = \frac{2}{n + 1},$$

kjer je n število dni v obdobju EMA.

- $EMA_{t-1,n}$: EMA iz prejšnjega dne.

RSI (Indeks relativne moči):

Indikator RSI (ang. "Relative Strength Index") meri hitrost in spremembo gibanja cen. Izračuna se kot razmerje med povprečnim dobičkom in povprečno izgubo v določenem obdobju. RSI se giblje med vrednostima 0 in 100, pri čemer RSI nad 70 pomeni prekuplenost (možen padec cene), RSI pod 30 pa pomeni preprodanost (možna rast cene). Pri trgovanju glede na indikator RSI zato v praksi velja, da delnico prodamo, ko RSI naraste nad 70, oz. jo kupimo, ko vrednost pade pod 30 [6].

Formula za izračun indikatorja RSI:

$$RSI_n = 100 - \frac{100}{1 + RS_n},$$

pri čemer je RS (Relative Strength) izračunan kot:

$$RS_n = \frac{\text{Povprečni dobički}_n}{\text{Povprečne izgube}_n},$$

kjer je:

- $\text{Povprečni dobički}_n$: povprečje pozitivnih donosov v obdobju n dni.
- $\text{Povprečne izgube}_n$: povprečje negativnih donosov v obdobju n dni.
- RS_n : razmerje med povprečnimi dobički in povprečnimi izgubami.

MFI (Indeks denarnega toka):

Indikator MFI (ang. "Money Flow Index") se izračuna podobno kot RSI, s tem da v formuli uporablja tudi podatke o obsegu trgovanja za prepoznavanje signalov prekupljenosti ali preprodanosti. MFI zavzame vrednosti med 0 in 100, kjer vrednosti nad 80 nakazujejo na prodajo, vrednosti pod 20 pa na nakup. V nekaterih primerih se uporabljajo tudi pragovi pri vrednostih 90 in 10 [13].

Formula za izračun MFI:

$$\text{MFI} = 100 - \left(\frac{100}{1 + \text{Razmerje denarnega toka}} \right),$$

pri čemer je:

$$\text{Razmerje denarnega toka} = \frac{\text{Pozitiven denarni tok}}{\text{Negativen denarni tok}}$$

in se Pozitiven denarni tok ter Negativen denarni tok izračunata po formuli:

$$\text{Denarni tok} = \text{Tipična cena} \times \text{Volumen},$$

$$\text{Tipična cena} = \frac{\text{Visoka cena} + \text{Nizka cena} + \text{Zaključna cena}}{3},$$

kjer je:

- Pozitiven denarni tok: seštevek vseh denarnih tokov za dni, ko je tipična cena višja od prejšnje tipične cene.
- Negativen denarni tok: seštevek vseh denarnih tokov za dni, ko je tipična cena nižja od prejšnje tipične cene.

MACD (Konvergenca in divergenca drsečih povprečij):

Indikator MACD (ang. "Moving Average Convergence Divergence") izračunamo kot razliko med 12-dnevnim in 26-dnevnim indikatorjem EMA. Za generiranje trgovalnih signalov potrebujemo vrednost MACD, signalno črto in MACD histogram. Signalno črto dobimo tako, da izračunamo 9-dnevni indikator EMA na vrednostih MACD. Zato trgovalne signale najpogosteje prikažemo s pomočjo MACD histograma, ki prikazuje razliko med MACD in signalno črto. Tako lahko sklepamo, da negativni trend izgublja moč, ko začne MACD histogram naraščati iz negativnih vrednosti proti 0. Ko se vrednost histograma dvigne nad 0, je to pogosto signal za nakup delnic. Podobno lahko sklepamo, da padanje MACD histograma iz pozitivnih vrednosti proti 0 nakazuje konec pozitivnega trenda. Ko vrednosti histograma padejo pod 0, je to pogosto signal za prodajo delnic [5].

Formula za izračun $MACD_{12,26}$:

$$MACD_{12,26} = EMA_{12} - EMA_{26}.$$

Formula za izračun signalne črte je sledeča:

$$\text{Signalna črta} = EMA_9(MACD_{12,26}).$$

MACD histogram pa se izračuna kot:

$$\text{Histogram} = MACD_{12,26} - \text{Signalna črta}.$$

Aroon:

Indikator Aroon meri, koliko časa je preteklo med posameznimi najvišjimi točkami in posameznimi najnižjimi točkami v določenem obdobju (običajno 25 dni). AroonUp meri, kako dolgo je minilo od zadnjega najvišjega vrha. Če je vrednost AroonUp blizu 100, to pomeni, da je bil nedavno dosežen najvišji vrh in da je rastoči trend močan. Če pa je AroonUp blizu 0, to kaže, da ni bilo nobenih pomembnih rasti, kar nakazuje šibek rastoči trend. AroonDown meri, kako dolgo je minilo od zadnje najnižje točke. Visoka vrednost AroonDown (blizu 100) pomeni, da je bilo nedavno doseženo najnižje dno, kar pomeni močan padajoči trend. Nizka vrednost AroonDown (blizu 0) kaže na šibek padajoči trend. Sekanje grafa AroonUp z grafom AroonDown služi kot trgovalni signal. Če AroonUp seka AroonDown od spodaj, to signalizira nakup. Podobno sekanje od zgoraj signalizira prodajo [12].

Formula za izračun indikatorja Aroon:

$$\text{AroonUp}(n) = \frac{n - \text{Število dni od najnižje cene}}{n} \times 100,$$

$$\text{AroonDown}(n) = \frac{n - \text{Število dni od najvišje cene}}{n} \times 100,$$

kjer je:

- n : število dni v obdobju.
- Število dni od najnižje cene: število dni, ki je preteklo od zadnje najnižje cene v obdobju.
- Število dni od najvišje cene: število dni, ki je preteklo od zadnje najvišje cene v obdobju.

H (Hurstov eksponent):

Indikator H omogoča analizo odstopanj od hipoteze učinkovitega trga EMH (ang. "Efficient Market Hypothesis"). Ta pravi, da so vse razpoložljive informacije že vključene v trenutne cene finančnih instrumentov, kar pomeni, da je težko ali nemogoče dosledno premagati trg z analizo preteklih podatkov ali javnih informacij. EMH vključuje tri stopnje učinkovitosti: šibko, srednjo in močno. Pri šibki obliki so pretekle cene že upoštevane, kar omejuje uporabnost tehnične analize, medtem ko srednja in močna oblika vključujeta še javne in zasebne informacije. Čeprav ta teorija omejuje uporabnost tehnične analize, mnogi vlagatelji menijo, da trgi niso vedno popolni, saj so ljudje pogosto iracionalni, kar ustvarja priložnosti za dobiček [11].

Hurstov eksponent je statistični indikator, ki meri stopnjo časovne avtokorelacije v časovni vrsti:

- **H < 0,5:** Cene kažejo povratno povezavo (ang. "mean-reversion"), kar pomeni, da se ponavadi vračajo k povprečni vrednosti. To nasprotuje EMH, ker kaže na določeno predvidljivost.
- **H ≈ 0,5:** Cena se obnaša kot popolnoma naključna, kar je v skladu z EMH. To pomeni, da ni vzorcev, ki bi jih lahko izkoristili za napovedovanje prihodnjih gibanj.

- **$H > 0,5$:** Nakazuje na trendno vedenje (persistenco), kjer pretekli podatki nakazujejo verjetno nadaljevanje trenda. To nasprotuje ideji EMH, saj pomeni, da trg ni popolnoma učinkovit in da so tržni vzorci lahko uporabni za napovedovanje.

V programskem jeziku Python lahko vrednosti indikatorjev tehnične analize računamo s pomočjo knjižnice `ta`, finančne podatke za izračun pa najpogosteje pridobimo s pomočjo knjižnice `yfinance`.

2.1 Naloge z namigi

Naloga 2.1 S pomočjo knjižnice `yfinance` pridobite podatke o cenah delnice AMZN od 1. 10. 2023 do 1. 10. 2024. Izpišite vrednost zaključne cene na dan 11. 1. 2024.

Naloga 2.2 Izpišite tudi najvišjo in najnižjo zaključno ceno v tem obdobju ter kdaj sta bili doseženi.

Naloga 2.3 S pomočjo knjižnice `matplotlib` narišite graf gibanja zaključnih cen v tem obdobju.

Naloga 2.4 Napišite metodo `SMAgraph(data)`, ki s pomočjo knjižnice `ta` na naboru podatkov `data` izračuna in nariše 50-dnevno in 200-dnevno preprosto drseče povprečje (SMA) glede na zaključne cene. Na grafu naj bo prikazano tudi gibanje zaključnih cen.

Namig: Uporabite funkcijo `trend.sma_indicator()`.

Naloga 2.5 Napišite metodo `EMAgraph(data)`, ki na naboru podatkov `data` izračuna in nariše 12-dnevni in 26-dnevni EMA glede na zaključne cene. Na grafu naj bo prikazano tudi gibanje zaključnih cen. Z zelenimi puščicami označite točke, kjer EMA signalizira nakup, z rdečimi pa točke, kjer EMA signalizira prodajo.

Namig: Uporabite funkcijo `trend.ema_indicator()`.

Naloga 2.6 S pomočjo knjižnice `ta` na naboru podatkov iz naloge 2.1 izračunajte vrednosti indikatorja RSI glede na zaključne cene. Izpišite vse dni, kjer se trgovalni signal "zadrži" spremeni bodisi v trgovalni signal "kupi" bodisi "prodaj".

Namig: Uporabite funkcijo `momentum.rsi()`.

Naloga 2.7 Napišite metodo `MFIgraph(data)`, ki bo izrisala graf MFI indikatorja.

Namig: Uporabite funkcijo `volume.MFIIndicator().money_flow_index()`.

Naloga 2.8 Napišite metodo `aroonGraph(data)`, ki s pomočjo knjižnice `ta` generira vrednosti `AroonUp` in `AroonDown`. Ločeno naj nariše tudi graf zaključnih cen in linijska diagrama `AroonUp` ter `AroonDown`.

Namig: Uporabite funkcije `trend.AroonIndicator()`, `aroon_up()` in `aroon_down()`.

Naloga 2.9 S pomočjo knjižnice `ta` na naboru podatkov iz naloge 2.1 generirajte

MACD trgovalne signale glede na zaključne cene.

Namig: Uporabite funkcije `trend.macd()`, `trend.macd_signal()` in `trend.macd_diff()`.

Naloga 2.10 Napišite metodo `MACDtrade(ticker, start, end, cash)`, ki bo simulirala trgovanje delnice `ticker` glede na indikator MACD v obdobju od `start` do `end` z denarnim vložkom `cash`. Metoda naj izpiše absolutni in relativni zaslužek ob koncu obdobja. Transakcijske stroške zanemarimo. Metodo testirajte na podatkih iz naloge 2.1 in denarnim vložkom 10.000 €.

Naloga 2.11 Za nabor podatkov iz naloge 2.1 izračunajte Hurstov eksponent. Glede na izračunan rezultat pojasnite, kakšna bi bila smiselna trgovalna strategija za ta finančni instrument.

Namig: Uporabite funkcijo `compute_Hc` iz knjižnice `hurst`.

2.2 Rešitve

Rešitev 2.1

```
import yfinance as yf

data = yf.download("AMZN", start='2023-10-01', end='2024-10-01')
print("Zaključna cena 11. 1. 2024 je bila: ",
      data.loc['2024-1-11', 'Close'].round(3), "$.")
```

Zaključna cena 11. 1. 2024 je bila: 155,18 \$.

Rešitev 2.2

```
highest_close = data['Close'].max().round(3)
lowest_close = data['Close'].min().round(3)

highest_close_date = data['Close'].idxmax()
lowest_close_date = data['Close'].idxmin()

print(f"Najvišja zaključna cena v tem obdobju: ",
      {highest_close}, "$ (dosežena ", {highest_close_date}, ")")
print(f"Najnižja zaključna cena v tem obdobju: ",
      {lowest_close}, "$ (dosežena ", {lowest_close_date}, ")")
```

Najvišja zaključna cena v tem obdobju: 200 \$ (dosežena 2024-07-02)

Najnižja zaključna cena v tem obdobju: 119,57 \$ (dosežena 2023-10-26)

Rešitev 2.3

```
import matplotlib.pyplot as plt

plt.plot(data['Close'], label='CLOSE', color='blue')
plt.grid()
plt.ylabel('Price ($)')
plt.show()
```



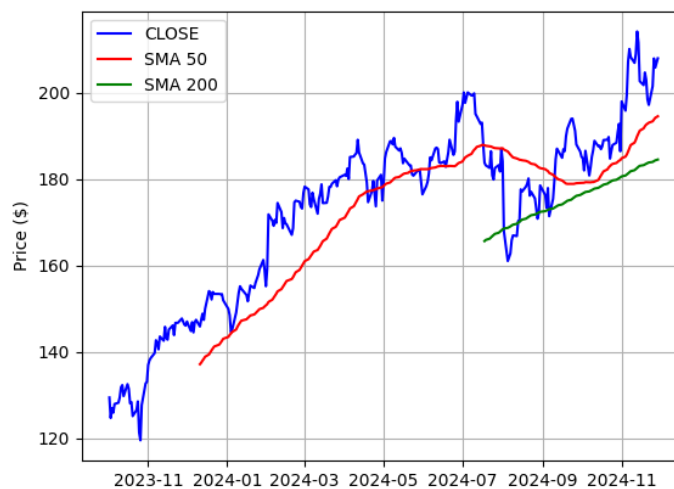
Slika 1: Gibanje zaključnih cen.

Rešitev 2.4

```
import ta
```

```
def SMAgraph(data):
    data['SMA50'] = ta.trend.sma_indicator(data['Close'],
    window=50)
    data['SMA200'] = ta.trend.sma_indicator(data['Close'],
    window=200)

    plt.plot(data['Close'], label='CLOSE', color='blue')
    plt.plot(data['SMA50'], label='SMA 50', color='red')
    plt.plot(data['SMA200'], label='SMA 200', color='green')
    plt.ylabel('Price ($)')
    plt.grid()
    plt.legend()
    plt.show()
```



Slika 2: Primer izrisa za podatke iz naloge 2.1.

Rešitev 2.5

```
import numpy as np

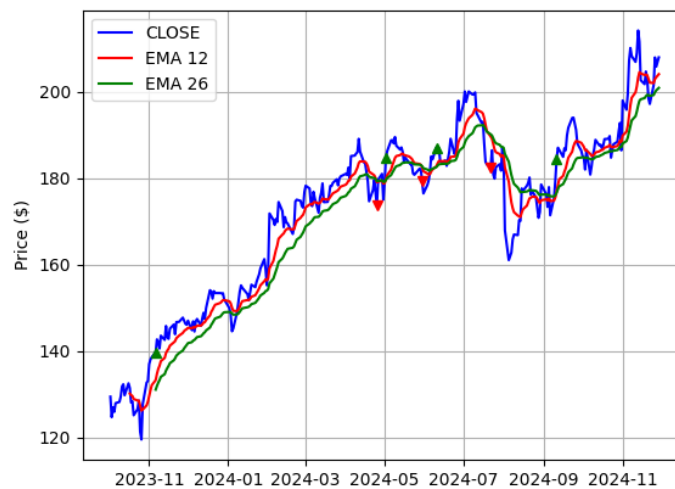
def EMAGraph(data):
    data['EMA12'] = ta.trend.ema_indicator(data['Close'], window=12)
    data['EMA26'] = ta.trend.ema_indicator(data['Close'], window=26)

    data['Bullish'] = 0
    data['Bullish'] = np.where(data['EMA12'] > data['EMA26'], 1, 0)
    data['Crossover'] = data['Bullish'].diff()

    plt.plot(data['Close'], label='CLOSE', color='blue')
    plt.plot(data['EMA12'], label='EMA 12', color='red')
    plt.plot(data['EMA26'], label='EMA 26', color='green')

    plt.plot(data.loc[data['Crossover'] == 1].index,
             data.Close[data['Crossover'] == 1], '^', color='green')
    plt.plot(data.loc[data['Crossover'] == -1].index,
             data.Close[data['Crossover'] == -1], 'v', color='red')

    plt.ylabel('Price ($)')
    plt.grid()
    plt.legend()
    plt.show()
```



Slika 3: Primer izrisa za podatke iz naloge 2.1.

Rešitev 2.6

```

data['RSI'] = ta.momentum.rsi(data['Close'])

buy_signals = []
sell_signals = []

for i in range(1, len(data)):
    rsi_new = data['RSI'].iloc[i]
    rsi_old = data['RSI'].iloc[i-1]

    if rsi_old >= 30 and rsi_new < 30:
        buy_signals.append(data.index[i])

    if rsi_old <= 70 and rsi_new > 70:
        sell_signals.append(data.index[i])

print("Trgovalni signal 'kupi':")
for i in buy_signals:
    print(i.date())

print("Trgovalni signal 'prodaj':")
for i in sell_signals:
    print(i.date())

```

Trgovalni signal 'kupi': 2024-08-05

Trgovalni signal 'prodaj': 2024-01-29, 2024-02-02, 2024-02-09, 2024-04-11, 2024-06-27

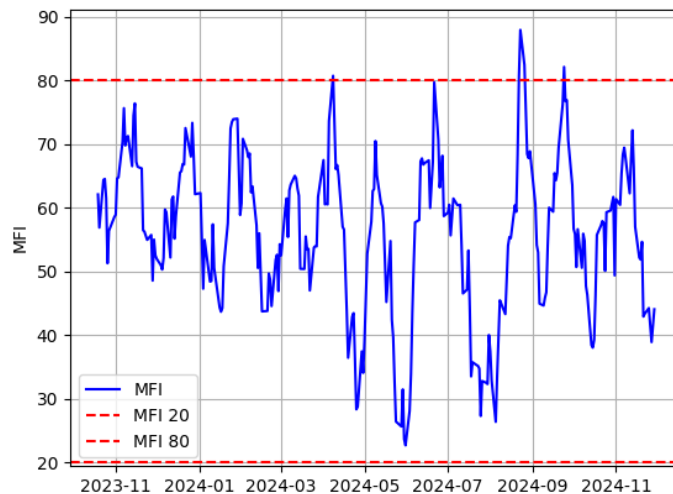
Rešitev 2.7

```

def MFIgraph(data):
    data['MFI'] = ta.volume.MFIIndicator(
        high=data['High'],
        low=data['Low'],
        close=data['Close'],
        volume=data['Volume'],
        window=14
    ).money_flow_index()

    plt.plot(data.index, data['MFI'], label='MFI', color='blue')
    plt.axhline(20, color='red', linestyle='--', label='MFI 20')
    plt.axhline(80, color='red', linestyle='--', label='MFI 80')
    plt.ylabel('MFI')
    plt.legend()
    plt.grid()
    plt.show()

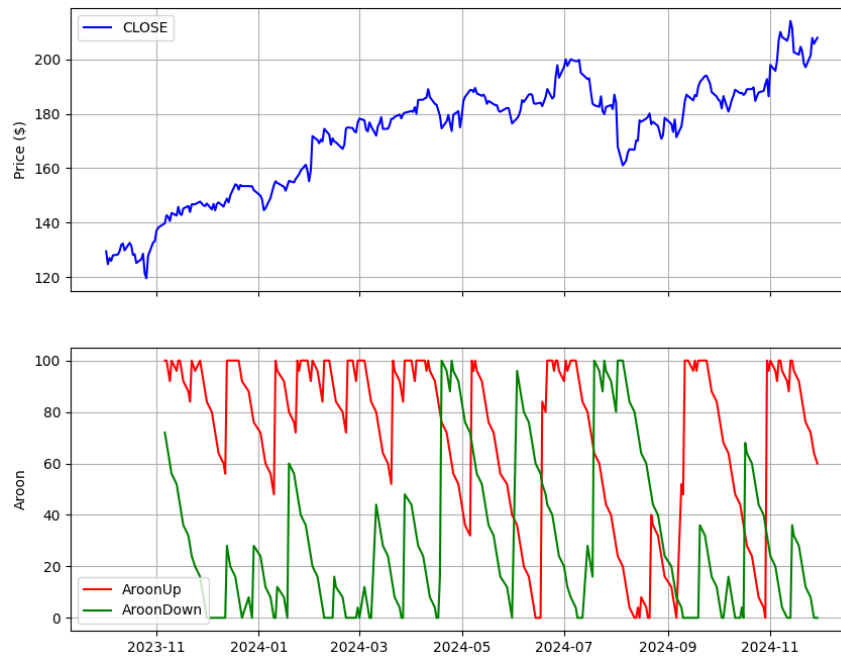
```



Slika 4: Primer izrisa za podatke iz naloge 2.1.

Rešitev 2.8

```
def aroonGraph(data):  
    aroon = ta.trend.AroonIndicator(high=data['High'], low=data['Low'],  
                                    window=25)  
  
    data['AroonUp'] = aroon.aroon_up()  
    data['AroonDown'] = aroon.aroon_down()  
  
    fig, axs = plt.subplots(2, 1, figsize=(10, 8), sharex=True)  
  
    axs[0].plot(data['Close'], label='CLOSE', color='blue')  
    axs[0].set_ylabel('Price ($)')  
    axs[0].grid()  
    axs[0].legend()  
  
    axs[1].plot(data['AroonUp'], label='AroonUp', color='red')  
    axs[1].plot(data['AroonDown'], label='AroonDown', color='green')  
    axs[1].set_ylabel('Aroon')  
    axs[1].grid()  
    axs[1].legend()  
  
    plt.show()
```



Slika 5: Primer izrisa za podatke iz naloge 2.1.

Rešitev 2.9

```
data['MACD_LINE'] = ta.trend.macd(data['Close'])  
data['MACD_SIGNAL'] = ta.trend.macd_signal(data['Close'])  
data['MACD_HIST'] = ta.trend.macd_diff(data['Close'])
```

Rešitev 2.10

```
def MACDtrade(ticker, start, end, cash):
    data = yf.download(ticker, start=start, end=end)

    data['MACD_LINE'] = ta.trend.macd(data['Close'])
    data['MACD_HIST'] = ta.trend.macd_diff(data['Close'])
    data['MACD_SIGNAL'] = ta.trend.macd_signal(data['Close'])

    data['Trade signal'] = np.where(data['MACD_HIST'] > 0, 1, -1)

    shares = 0
    startCash = cash
    for i in range(1, len(data)):
        price = data['Close'].iloc[i]
        if data['Trade signal'].iloc[i] == 1 and cash >= price:
            if shares != 0:
                newShares = int(cash // price)
                cash -= newShares * price
                shares += newShares
            else:
                shares = int(cash // price)
                cash -= shares * price

        elif data['Trade signal'].iloc[i] == -1 and shares != 0:
            cash += shares * price
            shares = 0

    if shares != 0:
        price = data['Close'].iloc[-1]
        cash += shares * price
        shares = 0

    profit = cash - startCash
    profit_percentage = (profit / startCash) * 100

    print(f"Končni zaslužek znaša {profit:.2f}$, kar je {profit_percentage:.2f}%")

MACDtrade('AMZN', '2023-10-01', '2024-10-01', 10000)
```

Končni zaslužek znaša 1268,86 \$, kar je 12,69 %.

Rešitev 2.11

```
from hurst import compute_Hc

H = compute_Hc(data['Close'], kind='price', simplified=True)[0]
```

Vrednost Hurstovega eksponenta znaša 0.752, kar nakazuje močno trendno obnašanje. To pomeni, da lahko na podlagi preteklih podatkov sklepamo, da se bo naložba tudi v prihodnje gibala v smeri trendov. Zato se bi pri trgovanju s to naložbo najbolj splačalo uporabiti indikatorje tehnične analize, ki zaznavajo trende na finančnih trgih.

3 Mere tveganja

V svetu financ se nenehno srečujemo z raznolikimi oblikami tveganja. Naložbe se med seboj razlikujejo po stopnjah tveganja in pričakovanih donosi. Vlagatelji za analizo podatkov in sprejemanje informiranih odločitev uporabljajo kazalnike oziroma mere tveganja. To so orodja, ki merijo, koliko donosa lahko pričakujemo glede na sprejeto tveganje [1].

V tem poglavju bomo predstavili nekatere najpogostejše uporabljene kazalnike.

Beta:

Finančna beta je mera volatilnosti oziroma občutljivosti finančnega instrumenta na gibanje cen trga. Beto izračunamo tako, da kovarianco donosov trga in donosov finančnega instrumenta delimo z varianco donosov trga v določenem obdobju. Za vrednosti donosov trga ponavadi vzamemo donose indeksa S&P 500, za katerega privzamemo, da ima beta vrednosti 1 [10].

Formula za izračun finančne bete:

$$\text{Beta} = \frac{\text{cov}(\text{Donosi trga}, \text{Donosi finančnega instrumenta})}{\text{var}(\text{Donosi trga})}.$$

Vrednost finančne bete nam o finančnem instrumentu pove sledeče:

- **Beta < 0:** Cene finančnega instrumenta se gibljejo v nasprotni smeri kot trg. Ko se cene na trgu dvigujejo, cena te naložbe običajno pada, in obratno, ko cene trga padajo, cena naložbe narašča.
- **Beta = 0:** Finančni instrument ni povezan s trgom, njegov donos ni odvisen od gibanja trga. To pomeni, da ta naložba ne bo doživela nihanj, ki bi jih povzročilo gibanje trga.
- **Beta < 1:** Finančni instrument je manj volatilen od trga. To pomeni, da se bo finančni instrument gibal počasneje od trga. Finančni instrument z beto 0,5 bo ob 5 % rasti trga tako pričakovano zrasel le za 2,5 %. Pri nakupu tega finančnega instrumenta je tveganje manjše, vendar je prav tako manjši tudi pričakovani donos.
- **Beta = 1:** Finančni instrument se giblje enako kot trg. Če bo vrednost trga zrasla za 5 %, lahko pričakujemo, da bo tudi vrednost finančnega instrumenta zrasla za 5 %. Tveganje je torej enako kot tveganje celotnega trga.
- **Beta > 1:** Finančni instrument je bolj volatilen od trga. To pomeni, da bodo nihanja cen finančnega instrumenta večja od nihanj na trgu. Finančni instrument z beto 1,5 bo tako ob 5 % rasti na trgu zrasel za 7,5 %. Pri nakupu tega finančnega instrumenta sprejmemo večje tveganje, vendar pričakujemo tudi večji donos.

Sharpeov količnik:

Sharpeov količnik meri donos neke naložbe glede na njeno tveganje. Z njim lahko izmerimo, koliko preseženega donosa lahko pričakujemo v primerjavi z naložbo brez tveganja (npr. državnimi obveznicami) [7].

Izračunamo ga po formuli:

$$S = \frac{R_p - R_f}{\sigma_p}.$$

Vrednost R_p predstavlja povprečni donos naložbe, R_f je povprečni donos netvegane naložbe, σ_p pa je standardni odklon donosov naložbe.

V praksi velja, da je naložba ustrezna, če je njen Sharpeov količnik večji od 1, saj prinaša glede na njeno tveganje visok preseženi donos. Tako seveda velja, da večja, kot je vrednost Sharpeovega količnika, bolj privlačna je naložba.

Posledično, če je Sharpeov količnik manjši od 1, to nakazuje, da naložba prinaša nizek donos glede na njeno tveganje. Naložba s Sharpeovim količnikom, enakim 0, pa ima ob upoštevanju tveganja enako donosnost kot netvegana naložba [7].

Traynorjevo merilo:

Traynorjevo merilo uporabljamo za merjenje uspešnosti naložbe glede na njeno sistematično tveganje. Z njim ocenimo, koliko preseženega donosa bo naložba ustvarila na enoto tveganja [9].

Traynorjevo merilo neke naložbe izračunamo po formuli:

$$T = \frac{R_p - R_f}{\beta_p}.$$

Vrednost R_p predstavlja povprečni donos naložbe, R_f je povprečni donos netvegane naložbe, β_p pa je finančna beta naložbe.

Traynorjevo merilo je zelo podobno Sharpeovemu količniku. Razlika med njima je le v tveganju, ki ga upoštevata. Traynorjevo merilo upošteva le sistematično tveganje naložbe, torej tveganje, ki ga prinaša celoten trg. Sharpeov količnik pa upošteva tudi nesistematično tveganje, v katerega je vključeno tveganje, ki ga prinaša posamezna naložba [9].

Traynorjev količnik uporabimo, kadar želimo primerjati portfelje z različnimi nivoji tveganja in ugotoviti, kateri nam bo prinesel najvišje donose glede na podano tveganje. Ko se odločamo za investicijo s pomočjo Traynorjevega količnika, izberemo portfelj z najvišjo vrednostjo [9].

3.1 Naloge z namigi

Naloga 3.1 Napišite metodo `beta(ticker, period, interval)`, ki izračuna finančno beto delnice `ticker` za obdobje `period` in za podatke pridobljene s frekvenco `interval` po prilagojeni ceni ob zaprtju. Nato izračunajte 5-letno finančno beto delnice AMZN na mesečnih podatkih. Za donose trga uporabite donose indeksa S&P 500.

Namig: Indeks S&P 500 ima oznako `^GSPC`.

Naloga 3.2 Na podlagi dnevnih podatkov izračunajte letni Sharpeov količnik indeksa S&P 500. Za netvegano naložbo uporabite donose ameriških kratkoročnih obveznic z oznako `^IRX`. Podatke izračunajte glede na ceno ob zaprtju.

Namig: V letu je 252 trgovalnih dni.

Naloga 3.3 Na podlagi dnevnihih podatkov izračunajte letno Traynerjevo merilo indeksa S&P 500 in delnice AMZN. Za netvegano naložbo uporabite donose ameriških kratkoročnih obveznic z oznako IRX. Podatke izračunajte glede na ceno ob zaprtju.

Naloga 3.4 Napišite metodo `MACDsharpe(ticker, cash)`, ki bo za izbrano delnico `ticker` in denarnim vložkom `cash` simulirala trgovanje glede na indikator MACD za obdobje enega leta. Metoda naj vrne vrednost letnega Sharpeovega količnika, izračunana na dnevnihih podatkih našega trgovanja.

Namig: Pomagajte si z nalogo 2.9.

3.2 Rešitve

Rešitev 3.1

```
def beta(ticker, period, interval):
    amzn = yf.download(ticker, period = period, interval=interval)
    gspc = yf.download("^GSPC", period = period, interval=interval)

    amzn['Returns'] = amzn['Adj Close'].pct_change()
    gspc['Returns'] = gspc['Adj Close'].pct_change()
    amzn = amzn.dropna()
    gspc = gspc.dropna()

    beta = np.cov(amzn['Returns'], gspc['Returns'])[0][1] /
            np.var(gspc['Returns'])
    return beta

beta("AMZN", '5y', '1mo')
```

Rešitev 3.2

```
from scipy.stats import gmean

data = yf.download("^GSPC", period='1y', interval='1d')['Close']
data.dropna(inplace=True)
returns = data.pct_change().mean()
std = data.pct_change().std()

irx = yf.download("^IRX", period='1y', interval='1d')['Close']
risk_free_yearly = gmean(irx) / 100
risk_free = (risk_free_yearly + 1) ** (1 / 252) - 1

sharpe_ratio = ((1 + returns - risk_free)**252 - 1)
                / (std * np.sqrt(252))
print(sharpe_ratio)
```

Rešitev 3.3

```

amzn = yf.download("AMZN", period='1y', interval='1d')['Close']
amzn.dropna(inplace=True)
sp500 = yf.download("^GSPC", period='1y', interval='1d')['Close']
sp500.dropna(inplace=True)
irx = yf.download("^IRX", period='1y', interval='1d')['Close']

amzn_returns = amzn.pct_change().mean()
sp500_returns = sp500.pct_change().mean()

risk_free_yearly = gmean(irx) / 100
risk_free = (1 + risk_free_yearly) ** (1 / 252) - 1

beta_amzn = beta("AMZN", '1y', '1d')
beta_sp500 = 1

treynor_amzn = ((1 + amzn_returns - risk_free)**252 - 1) /
               beta_amzn
treynor_sp500 = ((1 + sp500_returns - risk_free)**252 - 1) /
                beta_sp500

print(treynor_amzn, treynor_sp500)

```

Rešitev 3.4

```

def MACDsharpe(ticker, cash):
    data = yf.download(ticker, period='1y', interval='1d')
    data['MACD_HIST'] = ta.trend.macd_diff(data['Close'])

    data['Trade signal'] = np.where(data['MACD_HIST'] > 0, 1, -1)
    data['Cash'] = np.zeros(len(data))

    data.loc[data.index[0], 'Cash'] = cash
    shares = 0
    cash_invested = 0

    for i in range(1, len(data)):
        price = data['Close'].iloc[i]
        if data['Trade signal'].iloc[i] == 1:
            if cash >= price:
                if shares != 0:
                    newShares = int(cash // price)
                    cash -= newShares * price
                    shares += newShares
                else:
                    shares = int(cash // price)
                    cash -= shares * price
            cash_invested = shares * price
            data.loc[data.index[i], 'Cash'] = cash + cash_invested

        elif data['Trade signal'].iloc[i] == -1:
            if shares != 0:

```

```

        cash += shares * price
        shares = 0
        cash_invested = 0
        data.loc[data.index[i], 'Cash'] = cash

data.dropna(inplace=True)
returns = data['Cash'].pct_change().mean()
std = data['Cash'].pct_change().std()

irx = yf.download("^IRX", period='1y', interval='1d')['Close']
risk_free_yearly = gmean(irx) / 100
risk_free = (1 + risk_free_yearly) ** (1 / 252) - 1

sharpe_ratio = ((1 + returns - risk_free) ** 252 - 1) /
               (std * np.sqrt(252))
return sharpe_ratio

```

4 Zaključek

Analiza tehničnih indikatorjev in mer tveganja je ključna pri sprejemanju odločitev v svetu financ. Indikatorji SMA, EMA, RSI, MFI, MACD, Aroon in Hurstov eksponent pomagajo vlagateljem prepoznati trende in morebitne obrate na trgu. Po drugi strani mere tveganja, kot so beta, Sharpeov in Traynorjev količnik, omogočajo oceno tveganja naložb v primerjavi z njihovim pričakovanim donosom. Medtem ko tehnični indikatorji osvetljujejo tržne vzorce, mere tveganja analizirajo potencialne izgube in natančneje usmerjajo vlagatelje pri upravljanju tveganj.

Zahvala

Iskreno se zahvaljujemo dr. Dragu Bokalu za vso strokovno pomoč, podporo in dragocene nasvete, ki so pripomogli k oblikovanju tega članka.

Literatura

- [1] S. Strašek, T. Jagrič. (2008). Borzni trgi. Univerza v Mariboru, Ekonomsko-poslovna fakulteta, Inštitut za ekonomsko diagnozo in prognozo.
- [2] T. Jagrič. (2023). Borzni trgi [zapiski predavanj]. Univerza v Mariboru, Ekonomsko-poslovna fakulteta.
- [3] D. Fister. (2023). Borzni trgi [zapiski vaj]. Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko.
- [4] J. Chen. *What is EMA? How to Use Exponential Moving Average With Formula.* (2024). Pridobljeno iz: <https://www.investopedia.com/terms/e/ema.asp>
- [5] B. Dolan. *What Is MACD?.* (2024). Pridobljeno iz: <https://www.investopedia.com/terms/m/macd.asp>
- [6] J. Fernando. *Relative Strength Index (RSI) Indicator Explained With Formula.* (2024). Pridobljeno iz: <https://www.investopedia.com/terms/r/rsi.asp>
- [7] J. Fernando. *Sharpe Ratio: Definition, Formula, and Examples.* (2024). Pridobljeno iz: <https://www.investopedia.com/terms/s/sharperatio.asp>

- [8] A. Hayes. *Simple Moving Average (SMA): What It Is and the Formula*. (2024).
Pridobljeno iz:<https://www.investopedia.com/terms/s/sma.asp>
- [9] W. Kenton. *Treynor Ratio: What It Is, What It Shows, Formula To Calculate It*. (2024).
Pridobljeno iz:<https://www.investopedia.com/terms/t/treynorratio.asp>
- [10] W. Kenton. *What Beta Means for Investors*. (2024).
Pridobljeno iz:<https://www.investopedia.com/terms/b/beta.asp>
- [11] H. Kini. *Unlocking Market Insights: The Hurst Exponent of Momentum in Trading*. (2025).
Pridobljeno iz:<https://medium.com/@hrishikeshkini/unlocking-market-insights-the-hurst-exponent-of-momentum-in-trading-ff5678429d9f>
- [12] C. Mitchell. *Aroon Indicator: Formula, Calculations, Interpretation, Limits*. (2024).
Pridobljeno iz:<https://www.investopedia.com/terms/a/aroon.asp>
- [13] C. Mitchell. *Money Flow Index - MFI Definition and Uses*. (2020).
Pridobljeno iz:<https://www.investopedia.com/terms/m/mfi.asp>

Optimizacija simulacije trgovanja z delnicami z večnitenjem v programskem jeziku Python

Optimization of Stock Trading Simulation with Multithreading in Python

Maša Galun, Mia Molnar, Ema Smolič, Drago Bokal

Univerza v Mariboru, Fakulteta za naravoslovje in matematiko, Koroška cesta 160, 2000 Maribor

Povzetek

V tem članku bomo uporabili orodja za tehnično analizo delnic, ki smo jih spoznali v članku Tehnična analiza delnic: indikatorji in mere tveganja v programskem jeziku Python avtorjev Galun et al. Nato bomo razvili Monte Carlo simulacijo trgovanja z delnicami, ki temelji na indikatorju MACD. Pri tem bomo uporabili osnove večnitenja, ki jih obravnava članek Večnitne simulacije čustvenih stanj in modularni preizkusi v programskem jeziku Python avtorjev Galun et al., da pospešimo izvajanje kode. Simulacija bo preizkušena na delnicah iz indeksa S&P 500, rezultati pa bodo prikazani z animacijo, kar omogoča vizualizacijo izkupička pri trgovanju ob upoštevanju različnih scenarijev.

Ključne besede: delnice, trgovanje, simulacija, večnitenje, Python

Abstract

In this article, we will use the tools for technical analysis of stock that we explored in the article Technical Analysis of Stocks: Indicators and Risk Measures in Python by Galun et al. We will develop a Monte Carlo stock trading simulation based on the MACD indicator. Additionally, we will apply the basics of multithreading, as discussed in the article Multithreaded Simulations of Emotional States and Unit Testing in Python by Galun et al., to increase the execution speed of the code. The simulation will be tested on stocks from the S&P 500 index, and the results will be presented with an animation, allowing for a visual representation of trading outcomes under various scenarios.

Key words: stocks, trading, simulation, multithreading, Python

1 Uvod

Razumevanje tveganj in potencialnih dobičkov je ključno za oblikovanje uspešnih trgovalnih strategij. Monte Carlo simulacija, ki temelji na naključnem vzorčenju, je ena izmed najučinkovitejših metod za modeliranje negotovosti na finančnih trgih. Omogoča simulacijo možnih gibanj cen delnic, analizo tveganja in testiranje različnih strategij. V tem članku bomo Monte Carlo simulacijo uporabili za modeliranje trgovanja z delnicami, pri čemer bomo upoštevali trgovalne signale tehničnega indikatorja MACD. Simulacijo bomo preizkusili na delnicah iz indeksa S&P 500, rezultate pa prikazali z animacijo. Poleg tega bomo kodo prilagodili za večnitno izvajanje, kar omogoča vzporedno obdelavo nalog in večjo učinkovitost simulacij.

2 Monte Carlo simulacija

Monte Carlo simulacija je metoda, ki uporablja naključno vzorčenje za reševanje problemov ali modeliranje sistemov, ki vključujejo negotovost. V finančnem svetu se ta tehnika pogosto uporablja za analizo tveganja, modeliranje prihodnjih gibanj cen delnic in simulacijo različnih trgovalnih strategij.

V tem poglavju bomo Monte Carlo simulacijo uporabili za modeliranje potencialnih rezultatov trgovanja z delnicami. Osnova za to bo koda iz članka [1], kjer smo se naučili, kako izračunati tehnični indikator MACD. Ta indikator bo zdaj služil kot vstopna točka za naš model, ki bo s pomočjo naključnih scenarijev in ponovitev simuliral možne tržne izide. Indikator bomo preizkusili na delnicah iz indeksa S&P 500. Cilj je pridobiti boljši vpogled v tveganja in potencialne dobičke, ki jih ponuja trg. Končne rezultate bomo nato prikazali še z animacijo.

2.1 Naloge z namigi

Naloga 2.1 Generirajte seznam oznak delnic iz indeksa S&P 500.

Namig: Seznam oznak delnic lahko pridobite iz Wikipedie na strani o S&P 500.

Naloga 2.2 S pomočjo knjižnice `yfinance` pridobite podatke o cenah delnic iz indeksa S&P 500 od 1. 1. 2020 do 31. 12. 2023. Podatke shranite v Excel datoteko z imenom `sp500_data.xlsx`, kjer naj bodo podatki vsake delnice shranjeni v svojem listu.

Namig: Za generiranje Excel tabele uporabite `pd.ExcelWriter` iz knjižnice `pandas`.

Naloga 2.3 Napišite metodi `get_next_nearest_date(data, target_date)` in `get_previous_nearest_date(data, target_date)`, ki za parameter prejmeta nabor podatkov o cenah delnic `data` in datum `target_date`. Prva metoda za podan datum `target_date` v naboru podatkov `data` poišče najbližji datum, ki se v naboru podatkov pojavi za njim. Druga metoda pa poišče najbližji datum, ki se v naboru podatkov pojavi pred njim.

Naloga 2.4 Definirajte abstrakten razred `Indicator`, ki naj vsebuje inicializacijski konstruktor in razredno metodo `generateTradeSignals(self, data)`. Nato iz njega izpeljite razred `MACD`, ki ima lastnosti `buy` in `sell`. Lastnost `buy` pomeni vrednost indikatorja MACD, nad katero delnico kupimo, lastnost `sell` pa vrednost indikatorja MACD, pod katero delnico prodamo.

Naloga 2.5 Definirajte razred `Stock` z lastnostmi `cash`, `cashInvested` in `shares` ter razrednima funkcijama `buy(self, price)` in `sell(self, price)`. Lastnost `cash` predstavlja trenutno razpoložljiv denar, lastnost `cashInvested` predstavlja denar, ki je trenutno investiran v delnico, in lastnost `shares` predstavlja število delnic, ki jih imamo trenutno kupljene. Metodi `buy` in `sell` naj simulirata nakup oz. prodajo delnic po ceni `price`.

Naloga 2.6 Napišite metodo `trading(data, cash, indicator, start, end)`, ki simulira trgovanje delnice iz nabora podatkov `data` z denarnim vložkom `cash`

na podlagi trgovalnih signalov indikatorja `indicator`. Metoda naj vrne dobiček v odstotkih za dano časovno obdobje od `start` do `end`. Ceno delnice za posamezen dan določite naključno na intervalu med najnižjo in najvišjo ceno. Ta cena se nato uporabi za sprejemanje odločitev o nakupu ali prodaji delnic.

Naloga 2.7 Napišite metodo `MACDMonteCarlo(data, start, end, datesEnd)`, ki bo izvedla Monte Carlo simulacijo za analizo strategije trgovanja z indikatorjem MACD za delnico iz nabora podatkov `data`. Zadnji dan vsakega meseca naj bo podan v polju `datesEnd`. Pragovi za nakup se naj gibljejo od 0 do 1 s korakom 0,1, pragovi za prodajo pa od 0 do -1 s korakom -0,1. Za vsak par pragov se naj izvede 10 simulacij trgovanja, ki izračunajo dobiček za vsak mesec od datuma `start` do datuma `end`. Metoda naj za vsak mesec in vsak par pragov izračuna delež dobičkov, delež izgub ter povprečni donos. Rezultat je 3D matrika z imenom `squares`, ki prikazuje uspešnost strategije za različne kombinacije pragov skozi čas.

Naloga 2.8 Napišite metodo `getColors(squares)`, ki kot parameter prejme matriko `squares` iz naloge 2.7 in generira barvne vrednosti za vizualizacijo uspešnosti trgovalnih strategij. Za vsak mesec izračuna največji dobiček, nato pa za vsako kombinacijo pragov nakupa in prodaje določi barvo glede na delež izgub (rdeča), relativni donos glede na donose posameznega meseca (zelena) in razliko med deležem dobičkov in relativnim donosom meseca (modra). Barvna matrika omogoča vizualno analizo uspešnosti strategij skozi mesece. Rezultat si shranite v parameter `allColors`.

Naloga 2.9 Napišite metodo `color_matrix(allColors, squares, k, frames, ticker)`, ki kot parameter prejme matriki `squares` iz naloge 2.7 in `allColors` iz naloge 2.8. Parameter `ticker` predstavlja ime delnice. Metoda naj ustvari vizualizacijo 11×11 matrike, v kateri so prikazani vsi izidi trgovanja za k -ti mesec. Vsaka celica vsebuje barvo, ki jo določa RGB vrednost iz matrike `allColors`. Na vsaki celici se naj izpiše tudi, koliko je znašal dobiček pri trgovanju. Matriko shranite kot sliko in jo dodajte v seznam okvirjev `frames` za nadaljnjo uporabo v animaciji.

Naloga 2.10 Napišite metodo `color_matrices_and_animation(allColors, squares, ticker)`, ki ustvari animacijo v formatu GIF. Za vsak mesec izdela sliko s pomočjo funkcije `color_matrix()` iz naloge 2.9. Nato ta seznam slik shrani kot animirani GIF z imenom `ticker_MACD.gif`, kjer je dolžina trajanja vsakega okvirja 600 ms.

Naloga 2.11 Za delnico AMZN izdelajte matriko iz naloge 2.10 za obdobje od 1. 1. 2023 do 31. 12. 2023.

2.2 Rešitve

Rešitev 2.1

```
import pandas as pd

tickers = pd.read_html('https://en.wikipedia.org/wiki/
                        List_of_S%26P_500_companies')[0]
tickers = tickers['Symbol'].values
print(tickers)
```

Rešitev 2.2

```
import yfinance as yf

with pd.ExcelWriter("sp500_data.xlsx") as writer:
    for ticker in tickers:
        try:
            data = yf.download(ticker, start='2020-01-01',
                               end='2023-12-31')
            data.to_excel(writer, sheet_name=ticker)

        except Exception as e:
            print(f"Napaka pri prenosu podatkov za {ticker}: {e}")
```

Rešitev 2.3

```
def get_next_nearest_date(data, target_date):
    future_dates = data[data['Date'] >= target_date]
    nearest_date = future_dates['Date'].iloc[0]
    return str(nearest_date)

def get_previous_nearest_date(data, target_date):
    past_dates = data[data['Date'] <= target_date]
    nearest_date = past_dates['Date'].iloc[-1]
    return str(nearest_date)
```

Rešitev 2.4

```
import ta
import numpy as np

class Indicator:
    def __init__(self):
        pass
    def generateTradeSignals(self, data):
        pass

class MACD(Indicator):
    def __init__(self, buy, sell):
        self.buy = buy
        self.sell = sell
```

```
def generateTradeSignals(self, data):
    data['Indicator'] = ta.trend.macd_diff(data['Price'])
    conditions = [
        (data['Indicator'] <= self.sell),
        (data['Indicator'] > self.sell) &
        (data['Indicator'] <= self.buy),
        (data['Indicator'] > self.buy)]
    values = [-1, 0, 1]
    data['Trade signal'] = np.select(conditions, values)
```

Rešitev 2.5

```
class Stock:
    def __init__(self, cash):
        self.cash = cash
        self.cashInvested = 0
        self.shares = 0

    def buy(self, price):
        if self.cash >= price:
            if self.shares != 0:
                newShares = int(self.cash // price)
                self.cash -= newShares * price
                self.shares += newShares
            else:
                self.shares = int(self.cash // price)
                self.cash -= self.shares * price
        self.cashInvested = self.shares * price
        return self.cash + self.cashInvested

    def sell(self, price):
        if self.shares != 0:
            self.cash += self.shares * price
            self.shares = 0
            self.cashInvested = 0
        return self.cash
```

Rešitev 2.6

```
import math
```

```
def trading(data, cash, indicator, start, end):  
    # generiramo naključne cene  
    data['Price'] = data.apply(lambda row: np.random.uniform(  
        row['Low'], row['High']), axis=1)  
    indicator.generateTradeSignals(data)  
  
    idxStart = data.index[data['Date'] == start].tolist()[0]  
    idxEnd = data.index[data['Date'] == end].tolist()[0]  
    stock = Stock(cash)  
    startCash = stock.cash  
  
    data['Cash'] = np.zeros(len(data))  
    data.loc[data.index[idxStart], 'Cash'] = stock.cash  
  
    # trgujemo glede na trgovalne signale  
    for i in range(idxStart, idxEnd + 1):  
        if math.isnan(data['Trade signal'].iloc[i]):  
            data.loc[data.index[i], 'Cash'] = stock.cash  
            continue  
        price = data['Price'].iloc[i]  
        if data['Trade signal'].iloc[i] == 1:  
            data.loc[data.index[i], 'Cash'] = stock.buy(price)  
  
        elif data['Trade signal'].iloc[i] == -1:  
            data.loc[data.index[i], 'Cash'] = stock.sell(price)  
  
        else:  
            data.loc[data.index[i], 'Cash'] = stock.cash  
                + stock.cashInvested  
  
    profit = data.loc[data.index[idxEnd], 'Cash'] - startCash  
    profit_percentage = (profit / startCash) * 100  
    return profit_percentage
```

Rešitev 2.7

```
def MACDMonteCarlo(data, start, cash, end, datesEnd):
    start = get_next_nearest_date(data, start)
    end = get_previous_nearest_date(data, end)
    startCash = cash
    squares = np.zeros((11, 11, 12), dtype=object)
    for i in range(11):
        for j in range(11):
            # določimo meje indikatorja
            buy = i/10
            sell = (-1)*j/10
            sim = []
            # izvedemo 10 MC simulacij
            for _ in range(10):
                macd = MACD(buy, sell)
                trading(data, cash, macd, start, end)
                months = []
                # izračunamo profit vsakega meseca za eno MC simulacijo
                for date in datesEnd:
                    endMonth = get_previous_nearest_date(data, date)
                    idxEnd = data.index[data['Date'] ==
                                         endMonth].tolist()[0]
                    profit = (data.loc[data.index[idxEnd], 'Cash']
                              - startCash) / startCash * 100
                    months.append(profit)
                # združimo rezultate vseh MC simulacij
                sim.append(months) # 10 * 12 (en kvader)
            # za vsak mesec izračunamo delež dobička in izgube
            # ter povprečni donos
            for k in range(12):
                square = []
                for s in sim:
                    square.append(s[k])
                profits = np.sum(np.array(square) > 0)
                losses = len(square) - profits
                pctProfits = profits / len(square)
                pctLosses = losses / len(square)
                avgProfit = sum(square) / len(square)
                squares[i][j][k] = [pctProfits, pctLosses, avgProfit]
    return squares
```

Rešitev 2.8

```
def getColors(squares):
    colors = np.zeros((11, 11, 12), dtype=object)

    for i in range(12):
        returnsMonths = []
        for j in range(11):
            # izračunamo največji dobiček v mesecu
            for k in range(11):
                returnsMonths.append(squares[j][k][i][2])
            maxProfit = max(returnsMonths)

        for j in range(11):
            for k in range(11):
                red = squares[j][k][i][1]
                # če imamo izgubo
                if squares[j][k][i][2] <= 0:
                    green = 0
                    blue = 0
                # če imamo dobiček
                else:
                    green = squares[j][k][i][2]/maxProfit
                    blue = abs(squares[j][k][i][0]
                               - abs(squares[j][k][i][2]/maxProfit))
                colors[j][k][i] = [red, green, blue]
    return colors
allColors = getColors(squares)
```

Rešitev 2.9

Osnovo za našo implementacijo smo povzeli iz članka Sekvenčni diagrami in simulacije čustvenih stanj v programskem jeziku Python avtorjev Dimič et al. [3].

```
def color_matrix(allColors, squares, k, frames, ticker):
    fig, ax = plt.subplots(figsize=(8, 8))
    for i in range(11):
        for j in range(11):
            rgbColors = allColors[i][j][k]
            text = squares[i][j][k][2]
            # ustavrimo en kvadrat v matriki
            rect = mpatches.Rectangle([(i)/11, -(j)/11], 1 / 11,
                                      -1 / 11, ec="none",
                                      color=(rgbColors[0],
                                              rgbColors[1],
                                              rgbColors[2]))
            ax.add_patch(rect)
            ax.text((i) / 11 + 0.022, -(j+1) / 11 + 0.039,
                    str(text.round(2)))

    ax.set_xlim(0, 1)
    ax.set_ylim(0, -1)
    ax.invert_yaxis()
    ax.set_xlabel("Buy")
```

```

ax.set_ylabel("Sell")
plt.title("Month: " + str(k+1) + " " + "Stock: " + ticker)
plt.savefig(str(k) + '.frame.png')
new_frame = Image.open(str(k % 12) + '.frame.png')
frames.append(new_frame)
plt.close()

```

Rešitev 2.10

Osnovo za našo implementacijo smo povzeli iz članka Sekvenčni diagrami in simulacije čustvenih stanj v programskem jeziku Python avtorjev Dimič et al. [3].

```

def color_matrices_and_animation(allColors, squares, ticker):
    frames = []
    # izdelamo matriko za vsak mesec
    for i in range(12):
        color_matrix(allColors, squares, i, frames, ticker)
    frames[0].save(ticker + '_MACD.gif', format='GIF',
                  append_images=frames[1:], save_all=True,
                  duration=600, loop=0)

```

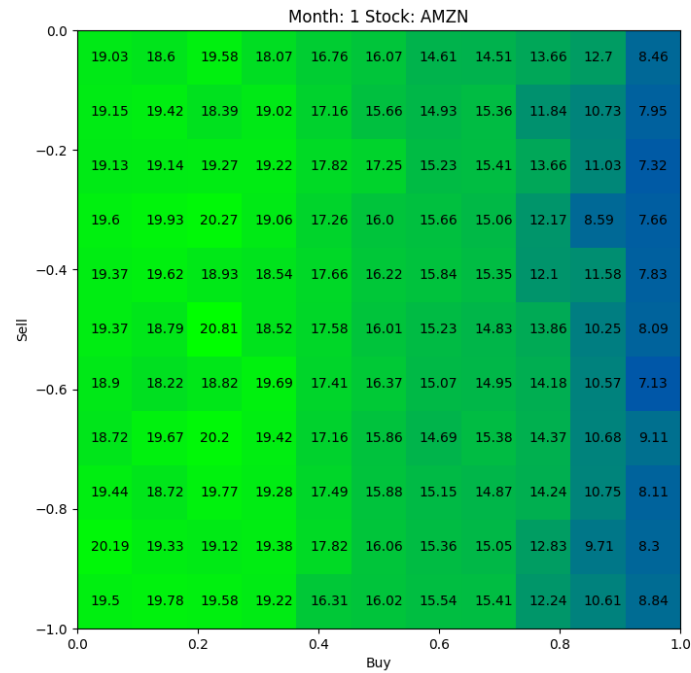
Rešitev 2.11

```

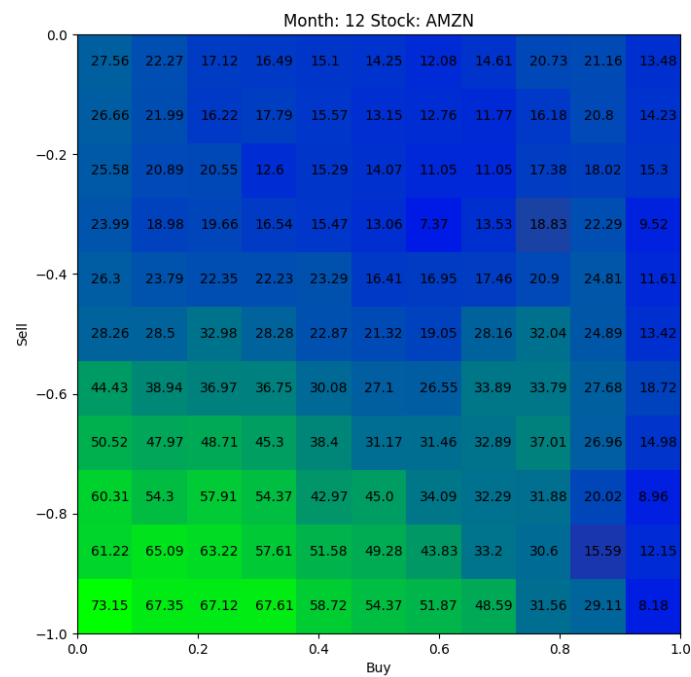
ticker = "AMZN"
data = pd.read_excel('sp500_data.xlsx', sheet_name=ticker)
start = '2023-01-01'
end = '2023-12-31'
datesEnd = ['2023-01-31', '2023-02-28', '2023-03-31',
            '2023-04-30', '2023-05-31', '2023-06-30',
            '2023-07-31', '2023-08-31', '2023-09-30',
            '2023-10-31', '2023-11-30', '2023-12-31']
cash = 10000
squares = MACDMonteCarlo(data, start, end, datesEnd, cash)
allColors = getColors(squares)
color_matrices_and_animation(allColors, squares, ticker)

```

Primer izrisa prve in zadnje matrike v animaciji:



Slika 1: Rezultati trgovanja delnice AMZN po prvem mesecu.



Slika 2: Rezultati trgovanja delnice AMZN po zadnjem mesecu.

3 Večnitna simulacija trgovanja

V tem poglavju bomo nadgradili programsko kodo iz prejšnjega poglavja tako, da jo bomo prilagodili za izvajanje v večnitnem načinu. Večnitenje omogoča, da se posamezni deli kode izvajajo vzporedno, kar bistveno poveča hitrost obdelave podatkov in učinkovitost simulacije. Učinkovitost izvajanja je še posebej pomembna pri kompleksnih nalogah, kot so Monte Carlo simulacije, kjer je potrebno obdelati veliko naključnih scenarijev.

Za komunikacijo med procesi večnitenja bomo uporabili blokirano komunikacijo, ki omogoča nadzorovan prenos podatkov med različnimi nitmi. Osrednjo vlogo bo igral proces nadzornik, ki bo koordiniral delo več delavcev. Nadzornik bo poskrbel za razdelitev nalog med delavce in zbiranje njihovih rezultatov. Te rezultate bo združil v celoto in z njimi izdelal animacijo v formatu GIF, ki bo ponazorila dinamiko trgovanja skozi različne scenarije. S to razdelitvijo dela bomo dosegli boljšo razporeditev obremenitve sistema in povečali hitrost simulacije. Za podrobnejšo razlago večnitenja priporočamo branje članka Večnitne simulacije čustvenih stanj in modularni preizkusi v programskem jeziku Python [2].

3.1 Naloge z namigi

Naloga 3.1 Vzpostavite okolje za delo z ukazom `MPI.COMM_WORLD` in pridobite informacijo o številu procesov z metodo `Get_rank()`.

Naloga 3.2 Napišite metodo `MonteCarlo(data, start, end, datesEnd, buy, sell)`, ki naredi 10 Monte Carlo simulacij trgovanja z indikatorjem MACD, ki ima mejo za nakup `buy` in mejo za prodajo `sell`. Za vsak mesec naj izračuna delež dobičkov, delež izgub ter povprečni donos in rezultate vrne v parameter `trade`.

Namig: Prilagodite metodo iz naloge 2.7.

Naloga 3.3 Napišite metodo `master(data, stock, start, end, datesEnd)`, ki bo predstavljala delo nadzornika večnitenja. Nadzornik naj delavcem pošlje `data`, `start`, `end`, `datesEnd`, `buy` in `sell`, pri čemer se `buy` giblje na intervalu od 0 do 1 s korakom 0,1, `sell` pa na intervalu 0 do -1 s korakom -0,1. Nadzornik od delavcev prejme podatke o trgovanju, ki si jih shrani v seznam `trades`. Nato v matriko `squares` shrani rezultate trgovanja za vsako celico matrike ($11 \times 11 \times 12$). Na koncu uporabi funkciji `getColors()` in `color_matrices_and_animation()` iz nalog 2.8 in 2.10, da generira animacijo trgovanja.

Naloga 3.4 Napišite metodo `worker()`, ki predstavlja delo ene niti. Od nadzornika naj pridobi parametre `data`, `start`, `end`, `datesEnd`, `buy` in `sell` ter jih uporabi na funkciji iz naloge 3.2. Izračunano naj pošlje nadzorniku.

Naloga 3.5 Kodo iz naloge 2.11 nadgradite z večnitenjem.

3.2 Rešitve

Rešitev 3.1

```
from mpi4py import MPI
```

```
comm = MPI.COMM_WORLD
rank = comm.Get_rank()
```

Rešitev 3.2

```
def MonteCarlo(data, start, end, datesEnd, buy, sell):
    startCash = 10000
    sim = []
    # izvedemo 10 MC simulacij
    for _ in range(10):
        macd = MACD(buy, sell)
        cash = startCash
        trading(data, cash, macd, start, end)
        months = []
        # izračunamo profit vsakega meseca za eno MC simulacijo
        for date in datesEnd:
            endMonth = get_previous_nearest_date(data, date)
            idxEnd = data.index[data['Date'] == endMonth].tolist()[0]
            profit = (data.loc[data.index[idxEnd], 'Cash']
                      - startCash) / startCash * 100
            months.append(profit)
        # združimo rezultate vseh MC simulacij
        sim.append(months)
    trade = []
    # za vsak mesec izračunamo delež dobička in izgube
    # ter povprečni donos
    for k in range(len(datesEnd)):
        square = []
        for s in sim:
            square.append(s[k])
        profits = np.sum(np.array(square) > 0)
        losses = len(square) - profits
        pctProfits = profits / len(square)
        pctLosses = losses / len(square)
        avgProfit = sum(square) / len(square)
        trade.append([pctProfits, pctLosses, avgProfit])
    return trade
```

Rešitev 3.3

```
def master(data, stock, start, end, datesEnd):
    squares = np.zeros((11, 11, 12), dtype=object)
    buy = 0.0
    sell = 0.0
    count_send = 1
    count_receive = 1
    trades = []
    while 1:
        for i in range(1, comm.size):
            # če smo že poslali vse kombinacije mej
            if count_send > 11*11:
                break

            # delavcu povemo, da pošiljanje še ni končano
            comm.send(False, i, 2)
            # delavcu pošljemo podatke
            comm.send([data, start, end, datesEnd,
                       buy, sell], i, 0)
            buy += 0.1
            if buy > 1:
                buy = 0
                sell -= 0.1
            count_send += 1
        for i in range(1, comm.size):
            if count_receive > 11*11:
                break
            # sprejmemo rezultate delavca
            trade = comm.recv(None, i, 1)
            trades.append(trade)
            count_receive += 1
        if count_receive > 11*11 and count_send > 11*11:
            break

    # vsem delavcem sporočimo, da je pošiljanje končano
    for i in range(1, comm.size):
        comm.send(True, i, 2)

    # shranimo podatke in izdelamo matriko
    for i in range(11):
        for j in range(11):
            trade = trades[i + 11*j]
            for k in range(len(datesEnd)):
                squares[i][j][k] = trade[k]
    colors = getColors(squares)
    color_matrices_and_animation(colors, squares, stock)
```

Rešitev 3.4

```

def worker():
    while 1:
        stop = comm.recv(None, 0, 2)
        # če je nadzornik sporočil, da je pošiljanje končano
        if stop:
            break

        # sprejmemo podatke nadzornika
        arr = comm.recv(None, 0, 0)
        data = arr[0].copy(deep=True)
        start = arr[1]
        end = arr[2]
        datesEnd = arr[3]
        buy = arr[4]
        sell = arr[5]
        squares = MonteCarlo(data, start, end, datesEnd, buy, sell)
        # rezultate simulacij pošljemo nadzorniku
        comm.send(squares, 0, 1)

```

Rešitev 3.5

```

start = '2023-01-01'
end = '2023-12-31'
datesEnd = ['2023-01-31', '2023-02-28', '2023-03-31', '2023-04-30',
            '2023-05-31', '2023-06-30', '2023-07-31', '2023-08-31',
            '2023-09-30', '2023-10-31', '2023-11-30', '2023-12-31']
ticker = "AMZN"
data = pd.read_excel('sp500_data.xlsx', sheet_name=ticker)

if rank == 0:
    master(data, ticker, start, end, datesEnd)
else:
    worker()

```

4 Zaključek

Monte Carlo simulacija se je izkazala za uporabno orodje pri modeliranju tveganja in analizi trgovalnih strategij. Z uporabo tehničnega indikatorja MACD smo ustvarili model, ki ponazarja potencialne izide trgovanja z delnicami. Implementacija večnitnega izvajanja je omogočila hitrejše in učinkovitejše obdelovanje veliko podatkov, kar je bistveno izboljšalo učinkovitost simulacij. S pridobljenimi rezultati in animacijami smo lahko bolje razumeli dinamiko trga in tveganja, povezanega s trgovanjem. Ta pristop odpira nove možnosti za nadaljnje raziskave in optimizacijo trgovalnih strategij, saj omogoča hitro prilagajanje modelov na različne tržne razmere. V prihodnosti bi bilo mogoče to metodo nadgraditi z dodatnimi indikatorji ali upoštevanjem drugih parametrov, ki vplivajo na uspešnost trgovanja.

Literatura

- [1] M. Galun, M. Molnar, E. Smolič, T. Jagrič, D. Fister. *Tehnična analiza delnic: indikatorji in mere tveganja v programskem jeziku Python*. Poslano v objavo v revijo Dianoia.
- [2] M. Galun, E. Smolič, D. Bokal. *Večnitne simulacije čustvenih stanj in modularni preizkusi v programskem jeziku Python*. Dianoia 7 (2023), 27-39.
- [3] J. Dimič, A. Lakner, K. Potočan, D. Bokal. *Sekvenčni diagrami in simulacije čustvenih stanj v programskem jeziku Python*. Dianoia 5 (2021), 53-69.
- [4] S. Strašek, T. Jagrič. (2008). *Borzni trgi*. Univerza v Mariboru, Ekonomsko-poslovna fakulteta, Inštitut za ekonomsko diagnozo in prognozo.

Crash course on hosting a website with focus on CORS

Hitri tečaj gostovanja spletne strani s poudarkom na CORS

Nikola Čokor, Jakob Battelino Prelog, Drago Bokal

DataBitLab d.o.o.

Fakulteta za elektrotehniko, računalništvo in informatiko Univerze v Mariboru, Koroška cesta 46, 2000 Maribor

Fakulteta za naravoslovje in matematiko Univerze v Mariboru, Koroška cesta 160, 2000 Maribor

Telekom Slovenije d.d.

Povzetek

Z eksplozijo interneta opazamo, da veliko več ljudi želi ustvariti svoja spletna mesta. Izdelava in gostovanje spletnega mesta je lahko za mnoge ljudi izziv, še posebej za tiste, ki niso ravno vešč računalništva in programiranja. To ustvarja tudi druge težave, med katerimi je glavna varnost. Ta članek vas nauči dobrih praks gostovanja spletnih strani. V prispevku se boste naučili, kako postaviti osnovno spletno mesto z uporabo `Node.js` in ga gostiti z uporabo Apache. Prispevek vas bo na koncu izzval tudi z dvema nalagama, da preverite svoje znanje.

Ključne besede: CORS, Apache, Linux, Node.js, VirtualBox

Abstract

With the explosion of the Internet, we see many more people wanting to create their own websites. Creating and hosting a website can be challenging for many people, especially those not well-versed in computers, programming, and Internet protocols. That creates other issues as well, one of which is security. This paper demonstrates some good website hosting practices. In the paper, you will learn how to set up a basic website using `Node.js` and host it using Apache. The paper will also challenge you with two problems at the end to test your knowledge.

Key words: CORS, Apache, Linux, Node.js, VirtualBox

1 Introduction

Today's world is run on computers and more importantly the Internet. Internet safety is one of the things a lot of people struggle to grasp and overcome. One of the most prevalent problems plaguing the early internet was Cross-Site Request Forgery (CSRF). CSRF is used to send fake requests to banks, the government, and other important places where it was used to steal money, information, etc. To solve this issue, a feature called Same-Origin Policy (SOP) was introduced. Unfortunately, this policy proved to be too inflexible for genuine requests. Since SOP was not good enough, Cross-Origin Resource Sharing (CORS) was introduced. They function pretty much the same, the only difference is that CORS provides a controlled mechanism for content requests from different origins. Due to these and other measures, CSRF is much less of an issue than it used to be.

E-mail naslovi: cokornikola@gmail.com (Nikola Čokor), jakobbp@gmail.com (Jakob Battelino Prelog), drago.bokal@um.si (Drago Bokal)

For this reason, in the following article, we will teach you how to set up a basic website and secure it. We will achieve this by using a technology called Apache HTTP Server (the current version is Apache2.4). Unfortunately, this article cannot cover everything but the basics. We still hope it will inspire you to learn further and start making your own websites.

2 Tasks

We structure our contribution as a series of tasks, each bringing you a step closer to hosting your own website. For the most confident and self-learning readers, the list of tasks may prove sufficient.

To encourage this self-learning, we have separated tasks from hints. This allows the reader to find relevant materials online. Should you fail to find satisfactory material, hints that are provided point to the material that was used in writing this article. These should suffice for moderately versed readers, who may just need a few pointers to reach the goal. And finally, for novice learners who are just looking for a tutorial, we provide the solutions for achieving the goal of hosting your own website.

2.1 Installing VirtualBox

For easier installation and use of Apache, we want to have a Linux system installed, since some people have Windows and maybe don't want a dual boot, this is the best option. Therefore as the first task, we will install VirtualBox. VirtualBox is just an Operating System emulator. Once VirtualBox has been installed you may progress. This task is not necessary, it is however recommended if you have Windows as it will be easier to progress further.

2.2 Installing Linux

In the case the previous task was ignored, the same can be done with this one.

For the ease of using Apache, we want to have Linux. In this task, this is exactly what we will do. We recommend Ubuntu (as that is what is going to be used in all subsequent tasks), but ultimately the choice is yours. Once Linux is installed, you are free to personalize it according to your wishes. If this is your first time using Linux, we recommend familiarizing yourself with some basic Linux commands. A really good resource for learning Linux basics is [1]. Solving the Grasshopper part should be enough, but you are also free to solve the other parts.

2.3 Installing Apache

We have finally arrived at the first task that cannot be ignored. That task is installing Apache on the Linux distribution. Apache is a web server we will use to host our website. After you have installed Apache, create a sample website to test that it is properly installed. The website can just be a simple Hello World! web page.

2.4 Allowing host access to VirtualBox

This is the last skippable task, all following tasks are unskippable. You only need to do this task if you did tasks 1 and 2. This task will allow us to access our server from our host machine (Windows).

2.5 Installing Node.js

Since we want to learn about proxies and want to simulate problems, we will need to run JavaScript on the server side. For that reason, we need to install `Node.js`. Once it is installed, create a simple `Hello Node!` webpage to test the installation.

2.6 Creating a Virtual Host

In this task, we create a virtual host for our website. A virtual host allows multiple websites to run on a single machine. For now, we won't be able to access the `Node.js` website from the Apache, this will be resolved in the following tasks. And we are redirected to the fallback website that is provided by Apache.

2.7 Proxy

In this task, configure a proxy to the `Node.js` website. With the proxy configured, the website is available right away through the browser. Accessing the website can be problematic when using Chrome, due to its default settings. We have found that the best browsers to access the website are either Firefox or Edge. You will not have any CORS issues accessing the website through the browser, CORS problems arise from fetching resources and other methods of accessing the website. For advanced insight into the interconnections of all the technologies used, we have prepared a more detailed website that allows you to experience CORS-related issues. This website can be found here: [2]. If you are not using the testing website, you will not be able to see the progress through the subsequent tasks. Only the `Test connection` button is used for testing the connection. You will get errors when trying to test the connection after adding your proxy, this is normal and will be fixed in the upcoming tasks.

2.8 Logging

If our server fails, sometimes it can be hard to diagnose the issues. For this reason, we use logging. Logging allows us to trace every action our server does. Therefore add logging to the virtual host. We recommend using at least the log level `error`, as it logs errors and critical tasks, though some log levels provide more details. This will not enable us to access the website, but it will help diagnose issues, that might arise later much easier.

2.9 Alias and Admin

A server alias allows Apache to redirect incoming requests to the correct server. We can also add multiple URLs if we wish. Setting an admin will allow us to receive emails about the status of our server and will notify end users to send reports to this email in case of any errors on their end. For this task, add at least one server alias and a server administrator. This doesn't fix our CORS issue either, it only provides another URL to use for access.

2.10 Access

There are several headers we can use to control access to our server by allowing, limiting, or restricting requests. These are used to protect your server from malicious attacks from the internet. The first header we will add is `Access-Control-Allow-Origin`, which allows us to control whether the response can be shared with the requesting origin. The second header we are adding is `Access-Control-Allow-Methods`, which allows us to control which methods we respond to. The final header we are adding is `Access-Control-Allow-Headers`, which allows us to dictate which HTTP headers are permitted in the request. The recommendation is not to use `*` in any header except `Access-Control-Allow-Origin`.

After successfully finishing this step, you will no longer encounter any errors connecting to the server using the `Test connection` button.

2.11 If statements

When the built-in functions are too slow or simply don't exist, we use if statements as a supplement. For this task, you will encase all the access headers into an if statement that checks whether the current time is within working hours (for example: between 8 and 16 o'clock) and grant access accordingly.

2.12 SSL certificate

Up to this point, our website is only accessible via `http`, which is insecure and primarily used for local resources. How do we make our website accessible via `https`? The process is both simple and complicated. It boils down to creating and adding SSL certificates to the website's configuration file.

Since our website needs to be accessible via `https` for upcoming tasks, we need to add the certificate in this task.

2.13 Preparation for the following tasks

The following tasks require some preparation.

In the `Node.js` server from Task 5, replace the simple `Hello Node!` with the `cors.html` provided in Task 7.

Last, but not least, create another SSL configuration and a `Node.js` server. This server will host the API (provided at the same location as `cors.html` in Task 7). Make sure you change the port to a different number (for example 3001), as two servers cannot share the same port.

Note, since we will be hosting the `cors.html` inside the `Node.js` server, every time we restart Apache, we also need to refresh the browser. Something we didn't need to do until now.

2.14 Issue 1

From now on, we will no longer learn new concepts through tasks, but through debugging on our own. This means that you will try to fix `Issue 1`. You are only permitted to edit the configuration files within `/etc/apache2/sites-available/`; all other files must remain unchanged. There will also be no hints, as you have learned everything you need to solve this issue in the previous tasks.

If you get an error similar to `ERR_CERT_AUTHORITY_INVALID` in this or any of the following tasks, simply access your websites, including the API website, through the specific browser once and click that you wish to proceed. This tells your browser that you trust that address.

2.15 Issue 2

This is very similar to the previous task. The only difference is that now you need to fix `Issue 2` in the `HTML` file. Note that if you haven't fixed the previous issue, you won't be able to fix this one.

2.16 Your issue

You will complete this task entirely on your own. What does this entail?

To start, add a new button to the `cors.html`. This button should have a corresponding JavaScript function that creates a request to the API website.

The second step is adding a response type to the API website. This is where you will simulate your issue.

Now, try to fix your issue by only editing the configuration file.

3 Hints

3.1 Installing VirtualBox

You can find the instructions here [3].

3.2 Installing Linux

You can find the instructions here [4].

3.3 Installing Apache

You can find the instructions here [5].

3.4 Allowing host access to VirtualBox

Look into different options for the network settings of your virtual host.

3.5 Installing Node.js

You can find the instructions here [6].

3.6 Creating a Virtual Host

Here is a simple example [7].

3.7 Proxy

Here is a simple example [8].

3.8 Logging

Here is a simple example [9].

3.9 Alias and Admin

You can find the instructions for Alias here [10].

You can find the instructions for Admin here: [11].

3.10 Access

Headers documentation can be found here [12].

We recommend you focus on CORS headers.

3.11 If statements

Documentation on the expressions for the Apache configuration files can be found here [13].

3.12 SSL certificate

You can find the instructions here [14].

4 Solutions

4.1 Installing VirtualBox

Go to [15]. Under platform packages, select your host (in our example, that is Windows host) and proceed with the download. Once downloaded, install VirtualBox.

4.2 Installing Linux

Download the latest Ubuntu image from [16].

After the image is downloaded, open VirtualBox and click on `New` to create a new virtual machine. Give your virtual machine a name and select its host folder and the ISO image we downloaded. Proceed with the next steps as recommended. Just make sure that when running the virtual machine for the first time, you select `Install Ubuntu`.

The actual Ubuntu installation might take some time.

4.3 Installing Apache

All you need to do to install Apache on Ubuntu is run the following commands in the terminal:

```
sudo apt update && sudo apt upgrade
```

```
sudo apt install apache2
```

But to use Apache, we also need to let it through the firewall. This is why we will also run the following:

```
sudo ufw allow 'Apache Full'
```

To test if we are successful, we will create a simple HTML file. First, navigate to `/var/www/html/`. There, execute the following command that will create a simple test file:

```
echo "<html><body><h1>Hello World!</h1></body></html>" |  
→ sudo tee test.html
```

Now we can access our test website by going to the browser and inputting `localhost/test.html`.

4.4 Allowing host access to VirtualBox

For this, we need to close the virtual machine if it is currently running. After it is closed, we can proceed to Settings -> Network and under Adapter 1, switch from NAT to Bridged Adapter. We can now access the website by inputting the virtual machine IP followed by the `/test.html` into the browser. For example: `192.168.138.17/test.html`.

To obtain the IP of your virtual machine, simply type the following in its console (not the Windows command prompt):

```
ifconfig | grep inet
```

4.5 Installing Node.js

Installing Node.js and creating a sample server is very easy. First, we install a package named `node` (and `npm` if not installed with `node`) with the following commands:

```
sudo apt install nodejs
```

```
sudo apt-get install npm
```

```
sudo apt install build-essential
```

After the packages `node` and `npm` are installed, we have to create our server/website. First, we create a directory, move inside it, and install our server with `npm`. While we can do this anywhere, we recommend you do not do it in the root, but rather in the user files, such as `Documents`. The creation commands are:

```
mkdir node_js
```

```
cd node_js
```

```
npm install express
```

The last thing we need is our `server.js` file and fill it with a basic setup, which is this:

```

1  const express = require("express");
2  const app = express();
3  const port = 3000;
4
5  app.get("/", (req, res) => {
6      res.send("Hello Node!");
7  });
8
9  app.listen(port, () => {
10     console.log(`Server running on port ${port}`);
11 });
12

```

Now all that is left is to run the server with:

```
npm start
```

and access it through the browser with `localhost:3000` for Linux and virtual machine IP for Windows (`192.168.138.17:3000`).

Remember, you need the Node.js server to be running if you wish to access it. So if it ever goes offline, don't forget to start it back up with `npm start`.

4.6 Creating a Virtual Host

First navigate to `/etc/apache2/sites-available/` and create a configuration file `cors.site.demo.conf`. Following is the most basic configuration file:

```

1  <VirtualHost *:80>
2      ServerName cors.site.demo
3  </VirtualHost>
4

```

But this is not enough for us to be able to see the website. We need to first run the following two commands:

```
sudo a2ensite cors.site.demo.conf
```

```
sudo a2dissite 000-default.conf
```

To finally be able to see the website, we need to navigate to `/etc` and add a new line to a file `hosts`. Copy the IP under `localhost`, increment the second to last number, and add the website name to the end. Example: my `localhost` is `127.0.0.1 localhost`, so my website is `127.0.2.1 cors.site.demo`.

Before we can access our website, we have to check if our configuration syntax is correct, which we do with the following:

```
sudo apachectl configtest
```

If the message says `Syntax OK`, that means that our configuration file is correct.

Finally, by restarting Apache using:

```
sudo systemctl restart apache2
```

and typing `cors.site.demo` into the browser, we can access the website.

To access the website from Windows, we also need to edit the `hosts` file on Windows. This file is located at `C:/Windows/System32/drivers/etc/hosts`. We edit it the same way we did the file on Linux, the only difference here is that we use the IP for the virtual machine we used in task 4, which would look like `192.168.138.17`.

Sadly, this alone will not allow us to see the website. The reason is that Apache doesn't know where our website is. The following tasks will allow us to access the website and its contents.

4.7 Proxy

To be able to access our website through the browser, we need to tell Apache where our website is. We will do this in our `test.com.conf` which is located in `etc/apache2/sites-available/`. In it, all we have to do is add 3 simple lines which are:

```
1 ProxyPreserveHost On
2 ProxyPass / http://localhost:3000/
3 ProxyPassReverse / http://localhost:3000/
4
```

This means your file should look as follows:

```
1 <VirtualHost *:80>
2     ServerName cors.site.demo
3
4     ProxyPreserveHost On
5     ProxyPass / http://localhost:3000/
6     ProxyPassReverse / http://localhost:3000/
7 </VirtualHost>
8
```

For the changes to take effect, you will need to check the syntax (make sure it says Syntax OK) and restart Apache:

```
sudo apachectl configtest
```

```
sudo systemctl restart apache2
```

This will finally allow us to see the contents of our website at the URL mentioned in the previous task.

But if we try to access it through the Test connection button, we get the following error (which is normal and will be fixed later):

```
Access to fetch at 'http://cors.site.demo/' from origin 'null'
has been blocked by CORS policy:
No 'Access-Control-Allow-Origin' header is present
on the requested resource.
```

4.8 Logging

To add logging to our website, simply navigate to `/etc/apache2/sites-available/` and add the following to `cors.site.demo.conf` from the previous task:

```
1 LogLevel trace6
2 ErrorLog ${APACHE_LOG_DIR}/test.error.log
3 CustomLog ${APACHE_LOG_DIR}/test.access.log combined
4
```

This means your file should look as follows:

```
1 <VirtualHost *:80>
2     ServerName cors.site.demo
3
4     LogLevel trace6
5     ErrorLog ${APACHE_LOG_DIR}/cors.site.error.log
6     CustomLog ${APACHE_LOG_DIR}/cors.site.access.log
7         ↪ combined
8
9     ProxyPreserveHost On
10    ProxyPass / http://localhost:3000/
11    ProxyPassReverse / http://localhost:3000/
12 </VirtualHost>
```

For the changes to take effect, you will need to check the syntax (make sure it says Syntax OK) and restart Apache:

```
sudo apachectl configtest
sudo systemctl restart apache2
```

4.9 Alias and Admin

To add an alias and an administrator, simply add the following to `cors.site.demo.conf`:

```
1 ServerAdmin admin@cors.site.demo
2 ServerAlias cors.site.com
3
```

This means your file should look as follows:

```
1 <VirtualHost *:80>
2     ServerAdmin admin@cors.site.demo
3     ServerName cors.site.demo
4     ServerAlias cors.site.com
5
6     LogLevel trace6
7     ErrorLog ${APACHE_LOG_DIR}/cors.site.error.log
8     CustomLog ${APACHE_LOG_DIR}/cors.site.access.log
9         ↪ combined
```

```

9
10     ProxyPreserveHost On
11     ProxyPass / http://localhost:3000/
12     ProxyPassReverse / http://localhost:3000/
13 </VirtualHost>
14

```

For the changes to take effect, you will need to check the syntax (make sure it says Syntax OK) and restart Apache:

```

sudo apachectl configtest

sudo systemctl restart apache2

```

If you want to access the website from one or more Alias you have defined, you will need to edit the `hosts` file on Linux and Windows.

To make things clear, the physical IP will stay the same, the only difference is that we will have multiple aliases we can use to access the website.

4.10 Access

The headers that we added look like the following:

```

1 Header always set Access-Control-Allow-Methods "POST, GET,
  ↳ OPTIONS"
2 Header always set Access-Control-Allow-Origin *
3 Header always set Access-Control-Allow-Headers
  ↳ "Keep-Alive, Content-Type, Origin"
4

```

This means your file should look as follows:

```

1 <VirtualHost *:80>
2     ServerAdmin admin@cors.site.demo
3     ServerName cors.site.demo
4     ServerAlias cors.site.com
5
6     LogLevel trace6
7     ErrorLog ${APACHE_LOG_DIR}/cors.site.error.log
8     CustomLog ${APACHE_LOG_DIR}/cors.site.access.log
  ↳ combined
9
10    ProxyPreserveHost On
11    ProxyPass / http://localhost:3000/
12    ProxyPassReverse / http://localhost:3000/
13
14    Header always set Access-Control-Allow-Methods "POST,
  ↳ GET, OPTIONS"

```

```

15     Header always set Access-Control-Allow-Origin *
16     Header always set Access-Control-Allow-Headers
        ↪ "Keep-Alive,Content-Type,Origin"
17 </VirtualHost>
18

```

For the changes to take effect, you will need to enable a header, check the syntax (make sure it says Syntax OK), and restart Apache:

```

sudo a2enmod headers

sudo apachectl configtest

sudo systemctl restart apache2

```

With this change, we can finally access our server without CORS-related issues.

4.11 If statements

To give API access only during working hours, we change the following:

```

1 Header always set Access-Control-Allow-Methods "POST, GET,
    ↪ OPTIONS"
2 Header always set Access-Control-Allow-Origin *
3 Header always set Access-Control-Allow-Headers
    ↪ "Keep-Alive,Content-Type,Origin"

```

into:

```

1 <If "%{TIME_HOUR} -gt 8 && %{TIME_HOUR} -lt 16">
2     Header always set Access-Control-Allow-Methods "POST,
    ↪ GET, OPTIONS"
3     Header always set Access-Control-Allow-Origin *
4     Header always set Access-Control-Allow-Headers
    ↪ "Keep-Alive,Content-Type,Origin"
5 </If>
6

```

This means your file should look as follows:

```

1 <VirtualHost *:80>
2     ServerAdmin admin@cors.site.demo
3     ServerName cors.site.demo
4     ServerAlias cors.site.com
5
6     LogLevel trace6
7     ErrorLog ${APACHE_LOG_DIR}/cors.site.error.log
8     CustomLog ${APACHE_LOG_DIR}/cors.site.access.log
    ↪ combined
9

```

```

10 ProxyPreserveHost On
11 ProxyPass / http://localhost:3000/
12 ProxyPassReverse / http://localhost:3000/
13
14 <If "%{TIME_HOUR} -gt 8 && %{TIME_HOUR} -lt 16">
15     Header always set Access-Control-Allow-Methods
16         ↪ "POST, GET, OPTIONS"
17     Header always set Access-Control-Allow-Origin *
18     Header always set Access-Control-Allow-Headers
19         ↪ "Keep-Alive, Content-Type, Origin"
20 </If>
</VirtualHost>

```

For the changes to take effect, you will need to check the syntax (make sure it says Syntax OK) and restart Apache:

```

sudo apachectl configtest

sudo systemctl restart apache2

```

4.12 SSL certificate

To make our website https accessible, we need to add certificates to our configuration file. But where do we get the certificates? For this example, we will use a resource called Certbot.

To use Certbot, we need to install it and its dependencies

```

sudo apt install python3 python3-venv libaugeas0

sudo apt-get remove certbot

sudo python3 -m venv /opt/certbot/

sudo /opt/certbot/bin/pip install --upgrade pip

sudo /opt/certbot/bin/pip install certbot certbot-apache

```

Now that the Certbot is installed along with its dependencies, we can use it to generate the certificates. Generating the certificates is very easy. The following commands will generate the certificate and automatically add it to the configuration file (when prompted for the email address, enter `admin@cors.site.demo`; and for redirection, choose option 2):

```

sudo ln -s /opt/certbot/bin/certbot /usr/bin/certbot

sudo certbot --apache

```

Now we simply deactivate the default SSL configuration and optionally deactivate the old http configuration with the following:

```
sudo a2dissite default-ssl.conf
sudo a2dissite cors.site.demo.conf
```

For the changes to take effect, you will need to check the syntax (make sure it says Syntax OK) and restart Apache:

```
sudo apachectl configtest

sudo systemctl restart apache2
```

Optional: Adding the auto-renewal on the certificate. Which is done with the following command:

```
echo "0 0,12 * * * root /opt/certbot/bin/python -c 'import
↳ random; import time; time.sleep(random.random() * 3600)'
↳ && sudo certbot renew -q" | sudo tee -a /etc/crontab >
↳ /dev/null
```

4.14 Issue 1

The solution is one single line that we add to the site configuration file. That one line is the following:

```
1 Header always set Content-Security-Policy
↳ "upgrade-insecure-requests"
2
```

For the changes to take effect, you will need to check the syntax (make sure it says Syntax OK) and restart Apache:

```
sudo apachectl configtest

sudo systemctl restart apache2
```

And that's it. All this does, is it upgrades all http requests to https.

4.15 Issue 2

The solution for this issue is rather easy, even though it might not seem like it. All we need to add to our API configuration file is the following:

```
1 <If "%{REQUEST_METHOD} == 'POST' && %{REQUEST_URI} ==
↳ '/issue2'">
2 Header always unset Access-Control-Allow-Origin
3 </If>
4
```

All this does is check for a request method and request URI. In the case both are matched, it removes the Access-Control-Allow-Origin header that was added by `Node.js`. The reason we check for the method is that the first request, that is sent, is a handshake request which just checks whether we have access or not to this resource. Since we have allowed access, the second request is a `POST` request, that is then forwarded to the `Node.js` website, which adds the header that needs to be removed. The reason we check the URI is so it does not interfere with the other issues.

5 Conclusion

Through these tasks, we have successfully shown how to build your first website and how to resolve some basic CORS errors. However, it should be noted that not all of the settings we used are secure. This article teaches users how to set up a simple Apache hosted website and a `Node.js` server. It does not teach users the most secure and optimal settings for public servers.

Anyone can use the tools provided and modify them as needed. These tools were created to help first-time users and those with more experience test their configuration files and resolve any errors.

Understanding how to create websites and improve their security contributes to a safer and better Internet. We hope that this article will inspire more people to create their own websites. Happy coding.

References

- [1] Linux Journey. "Learn the ways of Linux-fu, for free.", Linux Journey [Online]. Dosegljivo: <https://linuxjourney.com>. [1.10.2024]
- [2] niksacokica. (Oktober 2024). "Cors tutorial", GitHub [Online]. Dosegljivo: https://github.com/niksacokica/cors_tutorial. [1.10.2024]
- [3] abhinav96. "How to Install VirtualBox on Windows?", Geeks-forGeeks [Online]. Dosegljivo: <https://www.geeksforgeeks.org/how-to-install-virtualbox-on-windows/>. [20.9.2024]
- [4] Canonical Ubuntu. "How to run an Ubuntu Desktop virtual machine using Virtual-Box 7", Canonical Ubuntu [Online]. Dosegljivo: <https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox#1-overview>. [20.9.2024]
- [5] Gcore. (December 2023). "How to Set Up a Basic Web Server on Ubuntu", Gcore [Online]. Dosegljivo: <https://gcore.com/learning/setting-up-basic-web-server-ubuntu>. [20.9.2024]
- [6] B. Bearnese and K. Juell. (Julij 2018). "How To Set Up a Node.js Application for Production on Ubuntu 18.04", DigitalOcean [Online]. Dosegljivo: <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-18-04>. [20.9.2024]
- [7] Apache HTTP Server Project. "VirtualHost Examples", Apache HTTP Server Project [Online]. Dosegljivo: <https://httpd.apache.org/docs/2.4/vhosts/examples.html>. [20.9.2024]

- [8] R. Kumar. (Januar 2024) "How to Setup Apache Reverse Proxy for Node.js App", TecAdmin [Online]. Dosegljivo: <https://tecadmin.net/apache-frontend-proxy-nodejs/>. [20.9.2024]
- [9] Loggly. (2022). "Apache Logging Basics", Loggly [Online]. Dosegljivo: <https://www.loggly.com/ultimate-guide/apache-logging-basics/>. [20.9.2024]
- [10] Apache HTTP Server Project. "Name-based Virtual Host Support", Apache HTTP Server Project [Online]. Dosegljivo: <https://httpd.apache.org/docs/2.4/vhosts/name-based.html>. [20.9.2024]
- [11] Apache HTTP Server Project. "Apache Core Features", Apache HTTP Server Project [Online]. Dosegljivo: <https://httpd.apache.org/docs/2.4/mod/core.html#serveradmin>. [20.9.2024]
- [12] MDN Web Docs. "HTTP headers", MDN Web Docs. [Online]. Dosegljivo: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers>. [20.9.2024]
- [13] Apache HTTP Server Project. "Expressions in Apache HTTP Server", Apache HTTP Server Project [Online]. Dosegljivo: <https://httpd.apache.org/docs/current/expr.html>. [20.9.2024]
- [14] Electronic Frontier Foundation. "Certbot Instructions", Electronic Frontier Foundation [Online]. Dosegljivo: <https://certbot.eff.org/instructions?ws=apache&os=pip>. [9.1.2025]
- [15] Oracle VirtualBox. "Download VirtualBox", Oracle VirtualBox [Online]. Dosegljivo: <https://www.virtualbox.org/wiki/Downloads>. [10.9.2024]
- [16] Canonical Ubuntu. "Download Ubuntu Desktop", Canonical Ubuntu [Online]. Dosegljivo: <https://www.ubuntu.com/downloads/desktop>. [10.9.2024]

VABILO AVTORJEM

Dianoia (grško διάνοια) po Platonu označuje védenje, razmišljanje o modelih stvarnosti, o naravoslovno-matematičnih in tehničnih temah. Uporabljajo ga matematiki (modeliranje) in znanstveniki (formuliranje problema), inženirji (načrtovanje sistema). Opredeljuje kompetenco, proces ali rezultat diskurzivnega razmišljanja, za razliko od neposrednega razumevanja obravnavane tematike. Aristotel to védenje naprej razdeli na teoretično (episteme) in praktično (phronesis).

Dianoia po Platonu torej označuje vmesni nivo človeškega spoznanja, prehod od intuitivnih občutkov do najglobljega spoznanja dejanskosti. Tako je idealna oznaka za objave v pričujoči reviji, ki povezujejo teoretična, znanstvena izhodišča z njihovo uporabno namembnostjo. Študentje, avtorji teh člankov, ste na prehodu od učenja k delu, od teoretičnega h konkretnemu, ki vas bo pripeljalo do kruha, do dela, s katerim boste odigrali svojo vlogo v družbi. Na tem prehodu pa poleg znanja, ki ga ponuja redno izobraževanje, potrebujete tudi izkušnje s konkretnih izzivov in mehke kompetence sodelovanja v ekipah delodajalcev, k čemur vas spodbuja in vam pri tem pomaga revija Dianoia.

V reviji bomo objavljali poljudne in strokovne članke s področja naravoslovja, matematike ali znanosti, ki uporabljajo znanja teh področij. Ciljna publika bralcev so v prvi vrsti delodajalci, ki tovrstna znanja potrebujejo in želijo izvedeti, kaj je kdo zanimivega razmislil na njihovem področju. V drugi vrsti so ciljna publika študentje, ki iščejo zamisli za svojo poklicno pot in lahko v reviji najdejo navdih za lastna raziskovanja in iskanje stikov s trgom dela.

Za kakovost izdelkov bo skrbel uredniški odbor in uredniški svet, v katerih so vrhunski strokovnjaki, povezani s področji, ki jih revija obravnava. Članki bodo anonimno recenzirani, o objavi pa na podlagi recenzije odloča uredniški odbor. Priporočljivo je, da avtorji besedilo spremenijo v skladu s priporočili recenzentov in da popravljeni članek z utemeljitvijo sprejema ali zavrnitve sprememb ponovno pošljejo v pregled. Uredništvo lahko objavo članka zavrne, če vsebinsko ali po merilih kakovosti ne ustreza standardom revije, o čemer avtorje obvestimo v najkrajšem možnem času.

S prispevkom v reviji bodo avtorji spodbujali širjenje znanja s področja naravoslovja in matematike ter tehnike oziroma izobraževanja teh področij in svoje poglede prenašali na trg dela in na prihajajoče generacije.

NAVODILA AVTORJEM

Avtorje prosimo, da pri pripravi članka upoštevajo naslednja navodila.

Če je članek napisan v slovenščini, naj ima angleški prevod naslova, povzetka in ključnih besed. Veseli bomo tudi prispevkov v angleščini, ki pa morajo imeti naslov, razširjen povzetek v obsegu 300 – 400 besed in ključne besede v slovenščini. Ključnih besed naj bo do šest.

Prispevki naj bodo zanimivi za širši krog bralcev. Ključna je intuitivna predstavitev zamisli in rezultatov, podrobnosti pa lahko ostanejo prihranjene za morebitni znanstveni članek, ki bi bil nadgradnja članka, objavljenega v reviji Dianoia.

Članek naj vsebuje naslov, ime avtorja (avtorjev) in sedež ustanove, kjer avtor(ji) dela(jo). Sledi naj povzetek, z največ 150 besedami, seznam ključnih besed in besedilo, ki ne presega 3000 besed. Besedilo naj bo zapisano v urejevalniku besedil MS Word 2010 oz. kasnejši ali LaTeX in naj uporablja objavljeno predlogo. Slike in tabele morajo biti oštevilčene in imeti natančen opis, da jih lahko razumemo brez preostalega besedila. Slike v elektronski obliki naj bodo visoke kakovosti v formatu PNG ali JPEG.

Prispevek v PDF obliki pošljite na naslov dianoia@um.si z zadevo: »Za revijo Dianoia«. Če bo sprejet v objavo, vas bomo prosili za izvirno obliko prispevka.