

Igra Nim skozi matematiko in spodbujevano učenje

Raziskovalna naloga

Področje:

Matematika

Avtor:

Gregor BOKAL

7. razred

Mentorstvo in strokovni pregled:

mag. Ambrož Demšar

dr. Drago Bokal

Lektura:

Ljudmila Bokal

Šola:

Osnovna šola Alojzija Šuštarja

(Štula 23, Ljubljana - Šentvid)

Ljubljana 2021/22

Vsebina

Povzetek	4
Uvod	5
Teoretični del	6
Opis igre Nim.....	6
Igra v matematiki	6
Verjetnost	6
Frekvenca dogodka	7
Relativna frekvenca dogodka	7
Porazdelitev	8
Strategije igranja	10
Čiste strategije	10
Mešane strategije.....	10
Popolnoma mešane strategije.....	10
Strategije igre Nim	11
Naključna strategija.....	11
Zmagovalna strategija.....	11
Naključno zmagovalna strategija.....	12
Matematična indukcija	13
1. Primer - Gaussov način seštevanja zaporednih števil	13
2. Primer - število podmnožic v množici	14
Splošna pravila matematične indukcije	16
Dokazovanje zmagovalne strategije.....	16
Računanje razdalje.....	18
Razdalja na premici	18
Razdalja v ravnini.....	19
Razdalja v prostoru	21
Splošna razdalja	22
Računalniški igralci	23
Živi igralec.....	23
Naključni igralec.....	23
Naključni uteženi igralec	23
Strateški igralec	23
Strateški naključni igralec.....	23

Učljivi igralec	24
Eksperimentalni del	26
Programiranje igralcev in igre	26
Razred Igralec	26
Klasični igralci	27
Uteženi naključni igralec	27
Učljivi igralec	27
Igra in raziskava	28
Rezultati in razprava eksperimentalne raziskave	29
Diagrami deleža zmag	31
Strateški : strateški	31
Naključni : strateški	31
Graf razdalje do optimalne strategije	34
$n \times 3 + 1$	34
$n \times 3 + 2$, $n \times 3 + 3$	34
Manjša števila	35
Večja števila	35
Povprečno učenje	36
Učljivi in naključni igralec	36
Učljivi in strateški naključni igralec	36
Učljivi in strateški igralec	37
Zaključki	38
Viri in literatura	39

Povzetek

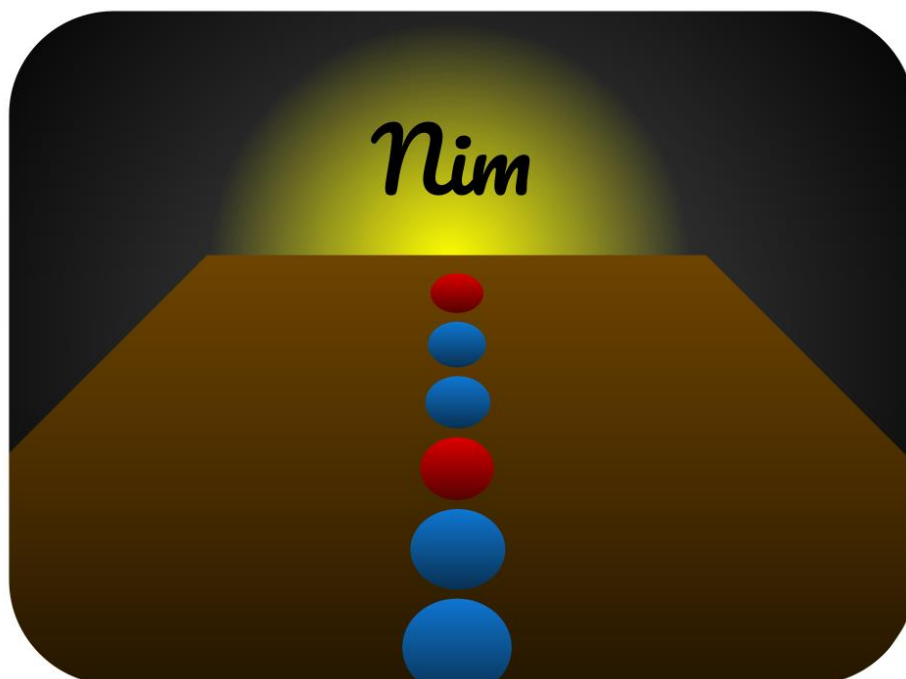
Pri matematični igri Nim imamo na polju nekaj žetonov. Ko smo na potezi, lahko vzamemo ali en ali dva žetona. Poraženec je tisti, ki vzame zadnji žeton. Na vsakem koraku se je treba odločati in vsaka odločitev je lahko usodna.

Namen matematike je, da nam olajša razumevanje pojavov. Tudi pri tej igri lahko z logičnim razmišljanjem pridemo do optimalne strategije. Kako pa bi do optimalne strategije prišel računalnik?

V naši raziskovalni nalogi v vlogi človeka z matematiko in logičnim razmišljanjem dokažemo optimalno strategijo, ki nas pripelje do zmage. Priložnost za učenje damo tudi računalniku, ki skozi odigrane igre pridobiva znanje na način spodbujevanega učenja. Na hitrost učenja vpliva tudi izbor soigralca. Če soigralec igra s strategijo, ki jo je izumil človek, računalnik razvije strategijo, s katero ga lahko premaga, kadar so okoliščine ugodne. Če pa soigralec igra naključno, potrebuje več časa za to, da pridobi enako strategijo.

Učenje računalnika spremljamo v za to narejenem Python programu. Med igro s simuliranim soigralcem se računalnik uči strategije s spodbujevanim učenjem. Med igranjem si zabeleži, koliko žetonov je vzel pri kateri situaciji. Če je po koncu igre poražen, zmanjša verjetnost, da kasneje spet izbere število žetonov, ki ga je uporabil. Obenem pa poveča verjetnost za poteze, ki je ni izbral. Če v igri zmaga, stori obratno.

Razlika med strategijo, ki jo je odkril človek, in strategijo, ki se jo je naučil računalnik, je, da človek razume, pod kakšnimi pogoji mora vzeti določeno število žetonov in kako mu to pride prav v primerih, ki jih še ni srečal. računalnik si pa le zapomni, koliko žetonov mora vzeti pri kolikem številu žetonov na polju, ki jih je že videl. Tega znanja pa ne zna posplošiti.



Slika 1: Igra Nim.

Uvod

V raziskovalni nalogi obravnavamo igro Nim. Ta na videz enostavna igra zna vzbuditi številna zanimiva matematična vprašanja in zamisli, na katera si želimo poiskati odgovore in razlage.

Seveda je vprašanje vsakega igralca, kako zmagati. Zmaga je namreč glavni cilj iger. Matematika nam tu lahko močno pomaga. V raziskavi tako s pomočjo matematike ugotavljamo, kako priti do *optimalne strategije* (tiste, s pomočjo katere lahko v ugodnih okoliščinah zmagamo). Igra Nim nam tukaj s svojo preprostostjo omogoča, da do nje pridemo brez potrebe po razumevanju prezapletenih stvari.

Res pa je tudi, da je cilj vsakega raziskovalca odkriti nekaj novega. Zato v naši raziskovalni nalogi ne ostajamo le pri iskanju strategij, kar pri igranju strateških družabnih iger počne praktično vsak, pač pa tudi iščemo odgovor na vprašanje, ali bi lahko optimalno strategijo našel tudi računalnik. Kako lahko to doseže? Za iskanje odgovorov na zastavljena vprašanja smo ustvarili Python program, v katerem računalnik simulira dva igralca, ki igro igrata.

Pod kakšnimi pogoji pa se računalnik hitreje uči? Takrat, ko igra z igralcem, ki se odloča naključno in ga lahko brez težav premaga, ali takrat, ko igra z igralcem, ki uporablja optimalno strategijo, ki smo jo poiskali mi. Predvidevamo, da se bo hitreje učil pri igri z igralcem, ki igra z našo strategijo. Je to res? Postavimo hipotezo in ugotovimo njeno resničnost v raziskovalni nalogi.

Očitno je, da je med človekovim in računalnikovim doumevanjem strategije velika razlika. Kakšna pa? Kdo strategijo bolje razume? Kdo jo zna bolje uporabiti? Zastavimo si cilj, da v raziskovalni nalogi odgovorimo tudi na ta vprašanja.

Teoretični del

Opis igre Nim

V naši raziskovalni nalogi obravnavamo matematično igro Nim. Pri njej je na polju nekaj žetonov. Recimo 10. Ko je eden od dveh igralcev na potezi, lahko vzame en žeton ali dva žetona. Poraženec je tisti, ki vzame zadnji žeton.

Igra v matematiki

Opisu igre sledi vprašanje, kaj sploh je igra. To besedo uporabljamo v številnih različnih pomenih. V literaturi najdemo definicijo za igre, s katerimi se lahko matematika ukvarja. V drugem delu uvoda knjige R. Jamnik (1973) piše, da je igra dogajanje, ki je odvisno od *igralcev* (udeležencev igre) in včasih od naključja. Igralci se morajo ravnati po natančno določenih pravilih in dogovorih. Igra poteka tako, da igralci v določenih fazah igre *izbirajo*, kar pomeni, da izmed dejanj, ki so jim na voljo, izberejo eno. Ponavadi tisto, za katero menijo, da bo za njih najugodnejša. Izbranemu dejanju rečemo *izbira* igralca. Faza, v kateri se izbiranje izvede, pa se imenuje *poteza*.

Verjetnost

Pri igrah so lahko nekateri dogodki bolj pričakovani kot drugi. V raziskovalni nalogi se veliko ukvarjamo s strategijami, kjer moramo vedeti tudi, s kakšno verjetnostjo naj v nekih okoliščinah vzamemo določeno število žetonov. Zato v tem poglavju s spletne strani (Teja Rajgelj (2002)) povzemimo razlago, kaj je verjetnost.

V vsakdanjem življenju skušamo za dogodke, ki jih ne moremo natančno napovedati, oceniti, s kakšno verjetnostjo se bodo zgodili. Pri izražanju mnenj pogosto uporabljamo izraze, kot so zagotovo, mogoče, možno, nemogoče itd. Podobno je tudi z verjetnostjo v matematiki. Verjetnost ugotovimo tako, da izvajamo *poskuse*. To so načrtne aktivnosti, ki jih izvajamo pod določenimi pogoji. To je npr. metanje igralne kocke in opazovanje števila pik, ki jih ta pokaže. Poskus po navadi izvedemo večkrat, da vidimo, kolikokrat se zgodi določen *dogodek*. Dogodek je vsak pojav, ki se v poskusu lahko zgodi, ali pa tudi ne. Dogodke ločujemo v tri skupine glede na to, kakšna je verjetnost, da se zgodijo. To so:

- **Gotovi dogodek** - ta se v poskusu vedno zgodi. Pri metanju kocke je to, na primer, da kocka pokaže vsaj 1 piko.
- **Slučajni dogodek** - ta se lahko v poskusu zgodi, lahko pa tudi ne. V primeru z igralno kocko je to, na primer, da pade število pik, ki je manjše od tri.
- **Nemogoči dogodek** - ta se nikoli ne zgodi. To je npr. da pri metanju kocke pade 7 pik.

Dogodke lahko ločimo tudi na:

- **Sestavljene dogodke** - takšni dogodki se lahko zgodijo na več načinov (npr. na kocki pade sodo število pik. To se lahko zgodi, da pade 2, 4 ali 6 pik).
- **Elementarne dogodke** - ti dogodki se zgodijo samo na en način (npr. kocka pokaže točno 5 pik).

Elementarnim dogodkom lahko rečemo tudi *izidi*.

Frekvenca dogodka

Ko večkrat izvedemo nek poskus, dobimo razne podatke, ki jih lahko obdelujemo. Oglejmo si primer, kjer izvajamo poskus, v katerem se lahko zgodita (elementarna) dogodka *A* in *B*. Po 100 izvedenih poskusih se je izid *A* zgodil 36 krat, izid *B* pa 64 krat. Ugotovimo lahko, da je *frekvenca* dogodka *A* enaka 36 in frekvenca dogodka *B* enaka 64. Frekvenca nam namreč pove točno število, kolikokrat se je v vseh poskusih dogodek zgodil. Pri tem je pomembno, da če seštejemo vse frekvence možnih izidov (elementarnih dogodkov) poskusa, dobimo število vseh poskusov, ki smo jih izvedli, kot pove naslednja trditev.

Trditev 1: Vsota vseh frekvenc izidov poskusa je enaka številu izvedenih poskusov.

Dokaz: Ena od pomembnih lastnosti izidov poskusa je, da se lahko hkrati zgodi natanko eden od njih. To pomeni, da se bo z vsakim poskusom za 1 povečala frekvenca enega od izidov. Če bomo torej sešteli vse frekvence le teh, dobimo število izvedenih poskusov.

Relativna frekvenca dogodka

V primerih, kot je ta, nas pogosto zanima, pri kolikšnem deležu poskusov se je zgodil kak dogodek. Če želimo pridobiti takšen podatek, moramo izračunati *relativno frekvenco*.

$$\text{Relativna frekvenca dogodka } A = \frac{\text{frekvenca dogodka } A}{\text{število izvedenih poskusov}}$$

Slika 2: Računanje relativne frekvence.

oz. $p = \frac{f}{n}$ (p = relativna frekvenca, f = frekvenca, n = število poskusov)

V našem primeru ni težko izračunati, da je relativna frekvenca dogodka *A* enaka 0,36 in relativna frekvenca dogodka *B* enaka 0,64.

Kadar poznamo dve od zgornjih količin, lahko vedno izračunamo tudi tretjo. Kako na podlagi frekvence in števila poskusov izračunamo relativno frekvenco, že vemo. Kako pa je, če sta nam znani frekvenca in relativna frekvenca, iščemo pa število poskusov?

Kot že vemo, nam relativna frekvenca pove, pri kolikšnem deležu poskusov se je zgodil izbran izid. Tako lahko ugotovimo, kolikšen delež vseh izvedenih poskusov predstavlja frekvenca. Če bomo torej frekvenco delili z relativno frekvenco, bomo dobili število vseh poskusov.

$$n = f : p$$

Slika 3: Računanje števila izvedenih poskusov.

Zdaj nam ostane le še primer, pri katerem iz relativne frekvence in števila poskusov izračunamo frekvenco. To pomeni, da vemo, kolikšna je celota in kolikšen del celote ustreza našim pogojem. Izračunati moramo, katero število nam del celote predstavlja. Če bi imeli pred seboj konkretne vrednosti, bi iskali na primer, koliko je $\frac{1}{2}$ od 50. Dobro vemo, da moramo, če želimo to izračunati, $\frac{1}{2}$ (relativno frekvenco) pomnožiti s 50 (številom poskusov). Tako bo veljalo, da frekvenco izračunamo tako, da relativno frekvenco množimo s številom vseh poskusov.

$$f = p \times n$$

Slika 4: Računanje frekvence.

Tokrat moramo izračunati, kolikokrat se je dogodek A zgodil v 30 izvedenih poskusih, če vemo, da je njegova relativna frekvenca 0,2. Če 0,2 pretvorimo v ulomek, dobimo $\frac{1}{5}$, torej moramo ugotoviti, koliko je petina od 30. $\frac{30}{5} = 6$. Tako na podlagi števila poskusov in relativne frekvence izračunamo, kolikokrat je bil poskus izveden.

Opazimo, da je zanimiva tudi vsota relativnih frekvenc. Če seštejemo vse relativne frekvence, dobimo rezultat enak 1. Se pravi:

Trditev 2: Vsota relativnih frekvenc vseh izidov poskusa je enaka 1.

Dokaz: Pri prejšnji trditvi smo dokazali, da je vsota vseh frekvenc izidov enaka številu poskusov. Iz frekvence pa računamo relativno frekvenco tako, da jo delimo s številom poskusov. Če torej velja naša prejšnja trditev, lahko tokrat trditev dokažemo kar z računom:

$$p_1 + p_2 + \dots + p_k = \frac{f_1 + f_2 + \dots + f_k}{n} = \frac{n}{n} = 1$$

Slika 5: Vsota relativnih frekvenc.

Porazdelitev

Zdaj ko poznamo osnovne pojme verjetnosti, jih uporabimo tako, da nam kasneje pridejo kar najbolj prav. Naš cilj je naučiti se, kako prikazati, s kakšnimi verjetnostmi bomo pri kakšnih situacijah izbrali določeno število žetonov. Pri igri Nim se lahko z vsako potezo zgodi eden od dveh dogodkov. To sta, da igralec vzame en žeton ali, da igralec vzame dva žetona. Recimo, da imamo igralca, ki igra povsem naključno. Pri njem bi bila torej pri velikem številu poskusov relativna frekvenca dogodka, pri katerem vzame 1 žeton, enaka 0,5 in prav taka pri dogodku, da vzame 2 žetona. To lahko opišemo s *porazdelitvijo*. Porazdelitev izidov poskusa ali krajše porazdelitev je nabor verjetnosti, ki nam pokažejo, kakšna je verjetnost vseh elementarnih dogodkov. Pri tem je treba upoštevati, da če se zgodi eden od teh izidov, se preostali ne morejo zgoditi. Zato je vsota vseh verjetnosti enaka 1. Porazdelitev se izraža npr. tako, kot na **sliki 6**.

$$\text{Porazdelitev pri metu kovanca} = \begin{pmatrix} C & G \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Slika 6: Porazdelitev pri metu kovanca.

Tu nam prvi element v množici pove, da je verjetnost dogodka, pri katerem pade prva stran kovanca (cifra), enaka $\frac{1}{2}$. Drugi element pa nam pove, da je taka tudi verjetnost, da pade druga stran kovanca (grb).

Če se držimo omenjenih pravil, lahko prikažemo porazdelitev pri metanju pravične igralne kocke s **sliko 7**.

$$\text{porazdelitev pri metanju igralne kocke} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}$$

Slika 7: Porazdelitev pri metanju igralne kocke.

Kako pa je pri seštevku pik, ko mečemo dve igralni kocki (to znanje zna priti zelo prav pri igri Catan)?

$$\text{porazdelitev pri metanju dveh kock} = \begin{pmatrix} 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ \frac{1}{36} & \frac{1}{18} & \frac{1}{12} & \frac{1}{9} & \frac{5}{36} & \frac{1}{6} & \frac{5}{36} & \frac{1}{9} & \frac{1}{12} & \frac{1}{18} & \frac{1}{36} \end{pmatrix}$$

Slika 8: Vsota pik pri metanju dveh kock.

Tukaj nam porazdelitev pove:

- Da sta verjetnost, da bo vsota pik 2, in verjetnost, da bo ta 12, enaki $\frac{1}{36}$. To je zato, ker se lahko zgodita le tako, da pade dvakrat po 1 pika (za vsoto 2), in dvakrat po 6 pik (za vsoto 12).
- Da sta verjetnosti za vsoto 3 in 11 enaki $\frac{1}{18}$, ker sta za obe možni dve kombinaciji pik (1 + 2, 2 + 1 in 5 + 6, 6 + 5).
- Da sta verjetnosti za vsoto 4 in 10 enaki $\frac{1}{12}$, ker so pri obeh možne 3 kombinacije pik (1 + 3, 2 + 2, 3 + 1 in 4 + 6, 5 + 5, 6 + 4).
- Da sta verjetnosti za vsoto 5 in 9 enaki $\frac{1}{9}$, ker so pri obeh možne 4 kombinacije pik (1 + 4, 2 + 3, 3 + 2, 4 + 1 in 3 + 6, 4 + 5, 5 + 4, 6 + 3).
- Da sta verjetnosti za vsoto 6 in 8 enaki $\frac{5}{36}$, ker je pri obeh možnih 5 kombinacij pik (1 + 5, 2 + 4, 3 + 3, 4 + 2, 5 + 1 in 2 + 6, 3 + 5, 4 + 4, 5 + 3, 6 + 2).
- Da je za verjetnost, da pade vsota 7 pik, enaka $\frac{1}{6}$, ker je zanjo možnih 6 kombinacij (1 + 6, 2 + 5, 3 + 4, 4 + 3, 5 + 2, 6 + 1).

Vrnimo se spet k igri Nim in igralcu, ki igra povsem naključno. Zdaj vemo, da bi lahko njegovo porazdelitev opisali s prikazom na **sliki 9**.

$$\text{porazdelitev pri } n \text{ žetonih } (n > 1) = \begin{pmatrix} 1 & 2 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Slika 9: Porazdelitev naključnega igralca.

Ker pa sta pri Nim - u vedno na voljo le dva izida, porazdelitev zapisujemo poenostavljeno:

$$\text{porazdelitev pri } n \text{ žetonih } (n > 1) = \left[\frac{1}{2}, \frac{1}{2} \right]$$

Prvi element pove verjetnost za izid, pri katerem bo vzeta 1 žeton, in drugi element za izid z 2 vzetima žetonoma.

Strategije igranja

V literaturi (Jamnik R. 1973) piše, da je strategija navodilo, ki ga uporabljamo pri igranju igre. Strategija določi, katero potezo naj uporabimo v različnih okoliščinah, oziroma, s kakšnimi verjetnostmi naj izberemo določeno potezo. Ker je načinov odločanja veliko, poznamo tudi več strategij.

Čiste strategije

Za čiste strategije je značilno, da nam pri potezi natančno povedo, kako naj se odločimo. Pri igri Nim se moramo vedno odločati ali vzamemo en, ali dva žetona. Strategija nam torej pove, kako se naj odločimo. Strategija **za eno potezo** je lahko npr. takšna:

$$\text{porazdelitev pri 5 žetonih} = [0, 1]$$

- Strategija nam pove, naj ne vzamemo enega žetona (v zgornjem primeru 0).
- Strategija zahteva, da vzamemo dva žetona (v zgornjem primeru verjetnost 1).

Mešane strategije

Pri mešanih strategijah je stvar podobna, le da nam včasih ne povedo, koliko žetonov naj vzamemo, ampak s kakšnimi verjetnostmi naj izberemo število vzetih žetonov. Pri tem ohranimo značilnost čistih strategij, da je vsota vseh verjetnosti 1. Te strategije ne moremo pokazati le v eni potezi, zato bo prikaz nekoliko drugačen:

$$\text{porazdelitev pri 1 žetonu} = [1, 0]$$

$$\text{porazdelitev pri 2 žetonih} = [\frac{4}{5}, \frac{1}{5}]$$

- Tokrat strategija želi, naj, ko ostane le še 1 žeton:
 - vsekakor vzamemo enega in
 - ne vzamemo dveh,
- ko pa ostaneta 2 žetona, naj:
 - z verjetnostjo $\frac{4}{5}$ (80%) vzamemo 1 žeton in
 - z verjetnostjo $\frac{1}{5}$ (20%) vzamemo 2 žetona.

Popolnoma mešane strategije

Popolnoma mešane strategije nam vedno povedo samo verjetnosti, s katerimi izbiramo potezo. Takih pri igri Nim sploh ni, saj pri enem žetonu ne moremo vzeti dveh. Vseeno pa lahko podamo primer za višje število žetonov:

$$\text{porazdelitev pri 8 žetonih} = [\frac{3}{4}, \frac{1}{4}]$$

- Ta strategija pove, da je verjetnost, da vzamemo en žeton $\frac{3}{4}$ (75%),
- verjetnost, da vzamemo dva žetona, pa je $\frac{1}{4}$ (25%).

Strategije igre Nim

V nadaljevanju sledi opis strategij, ki jih uporabljajo računalnikovi soigralci. To so:

Naključna strategija

Naključna strategija je **mešana** strategija, ki ima za vsako potezo porazdelitev $[\frac{1}{2}, \frac{1}{2}]$, razen v primeru, kjer je na polju le še en žeton.

Zmagovalna strategija

Ta strategija je **čista**. Za vsako število žetonov pove, koliko naj jih vzamemo. Igralca, ki jo uporablja, v ugodnih okoliščinah pripelje do zmage. Če je v dani situaciji nemogoče zmagati, se lahko uporabnik zmagovalne strategije odloči za katero koli potezo. Da preprečimo zmedo in ostanemo pri čisti strategiji, recimo, da vzame 1 žeton. Razmislimo zdaj, kakšna mora biti porazdelitev pri zmagovalni strategiji. Začnimo kar na koncu, saj je rezultat igre odvisen od zadnje poteze.

Hitro lahko ugotovimo, da potem, ko ostane le še en žeton, zmaga igralec, ki ni na potezi, saj igralec, ki je na potezi, nima druge izbire, kot da vzame zadnjega.

Ko ostaneta na polju dva žetona, zmaga igralec, ki je na potezi. Ko vzame en žeton, za njim ostane le še eden, ki ga vzame drugi igralec.

porazdelitev pri 2 žetonih = [1, 0]

Ko ostanejo trije žetoni, prav tako zmaga igralec, ki je na potezi, a le, če vzame dva žetona. Tako zopet ostane le še eden, ki ga pobere nasprotnik.

porazdelitev pri 3 žetonih = [0, 1]

Ko ostanejo štirje žetoni, zmaga igralec, ki ni na potezi. Če igralec, ki je na vrsti, vzame en žeton, nasprotnik zmaga (če igra tako, kot je zapisano v odstavku, kjer ostanejo trije žetoni). Če pa vzame dva žetona, ostaneta po njegovi potezi le še dva. V tem primeru nasprotnik vzame en žeton, zadnjega pa vzame prvi igralec.

Ko ostane pet žetonov, zmaga igralec, ki je na vrsti, saj bodo z odstranitvijo enega žetona ostali štirje. V primeru, ko ostanejo štirje žetoni, pa že vemo, da zmaga igralec, ki ni na vrsti.

porazdelitev pri 5 žetonih = [1, 0]

Ko ostane šest žetonov, je stvar enaka, le da mora igralec, ki je na potezi, vzeti dva žetona.

porazdelitev pri 6 žetonih = [0, 1]

Noben igralec, ki želi zmagati, si ne želi, da je na začetku njegove poteze na polju sedem žetonov. Če vzame enega, nasprotnik vzame dva, in ostanejo štirje žetoni. Mi pa že dobro vemo, da, kjer so štirje žetoni, tam je poraz. Če pa prvi igralec vzame dva žetona, nasprotnik vzame enega in tako je situacija enaka.

V primeru, da ostane osem ali devet žetonov, igralec zmaga tako, da vzame toliko žetonov, da jih naslednjemu ostane sedem.

porazdelitev pri 8 žetonih = $[1, 0]$ porazdelitev pri 9 žetonih = $[0, 1]$

Ko je v igri na začetku 10 žetonov, ne glede na to, ali igralec vzame en ali dva žetona, jih ostane devet ali osem. V tem primeru je že znano, da igralec, ki je takrat na vrsti, zmaga. Zato igralec, ki igre ne začne, le ne sme zavoziti in njegova zmaga je zagotovljena.

Preglednica 1: prikaz optimalne strategije in možnih situacij.

Število žetonov, ki jih vidimo na začetku poteze	Število žetonov, ki jih moramo vzeti	Ali lahko zmagamo proti pametnemu igralcu?
10	?	×
9	2	✓
8	1	✓
7	?	×
6	2	✓
5	1	✓
4	?	×
3	2	✓
2	1	✓
1	?	×

Če uporabljamo ta pravila odzemanja in igramo optimalno, ne glede na število žetonov vedno velja, da če na polju ostane število žetonov, ki je za ena večje od večkratnika števila tri (vključno z ena) oz. ga lahko opišemo z izrazom $3 \cdot n + 1$, zmaga igralec, ki ni na vrsti. Vsa omenjena števila žetonov, kjer zmaga igralec, ki ni na potezi, so namreč za tri večja od prejšnjega takšnega števila. To pomeni, da mora igralec, če je le mogoče, vzeti toliko žetonov, da nasprotniku ostane število žetonov, ki vodi v poraz.

Naključno zmagovalna strategija

Ta **mešana** strategija igralca pod ugodnimi okoliščinami pripelje do zmage z enakimi potezami, kot zmagovalna strategija. Če je situacija videti nemogoča, pa se odloča naključno $[\frac{1}{2}, \frac{1}{2}]$.

Matematična indukcija

V naši raziskovalni nalogi obdelujemo številne zanimive domneve. Prav bi prišlo, če bi jih tudi dokazali. Način, s katerim lahko storimo kaj takega, je opisan v literaturi Kavka D., Pavlič G., Rugelj M., Šparovec J. (2016), in sicer se imenuje *matematična indukcija*. Matematična indukcija je orodje, s katerim matematiki dokazujejo, da neka trditev velja za vsa naravna števila. Z njo lahko tudi potrdimo zmagovalno strategijo. Za boljše razumevanje, jo najprej uporabimo pri nekoliko lažjih primerih:

1. Primer - Gaussov način seštevanja zaporednih števil

Znana je anekdota, da je matematik Gauss že zelo mlad na zelo hiter način izračunal vsoto števil od 1 do 100. Ugotovil je, da dasta prvo in zadnje število (1 in 100) vsoto 101. Isto naredijo tudi 2 in 99 ter 3 in 98... in tako naprej. Takih kombinacij je 50 ($100 : 2$), zato je 101 pomnožil s 50 in dobil rezultat 5050. Ali je ta način reševanja ustrezen za vsa števila, lahko preverimo z matematično indukcijo.

Domneva, ki jo dokazujemo, je: $1 + 2 + 3 + \dots + n = (1 + n) \cdot (n : 2)$
(n pomeni število, vsoto števil, do katerega iščemo. Desna stran enačbe predstavlja del računa, ko v osnovnem primeru 101 pomnožimo s 50.)

Trditev: Vsota vseh naravnih števil do n je enaka $(1 + n) \times (n : 2)$.

Dokaz: Dokazovanje začnemo na prvem / prvih nekaj naravnih številih ($n = 1$). V tem primeru to pomeni, da moramo sešteti vsa naravna števila od 1 do 1.

$$\begin{aligned}(1 + n) \times (n : 2) &= \\= (1 + 1) \times (1 : 2) &= \\= 2 \times 0,5 &= \\= 1\end{aligned}$$

Dobimo rezultat 1, torej je v tem primeru domneva potrjena. Zdaj dokažimo domnevo še na nekaj naslednjih številih ($n = 2$):

$$\begin{aligned}(1 + n) \times (n : 2) &= \\= (1 + 2) \times (2 : 2) &= \\= 3 \times 1 &= \\= 3\end{aligned}$$

Tokrat izraz da rezultat 3. Ker je vsota števil od 1 do 2 enaka 3, smo uspeli domnevo dokazati tudi pri $n = 2$. Dokazovanje lahko nadaljujemo v neskončnost, zato je naslednji korak pri matematični indukciji, da dokažemo *indukcijski korak*. To pomeni, da predpostavimo, da trditev velja za n in vsa predhodna števila in nato dokažemo, da velja tudi za $n + 1$.

Formula za $n + 1$ je taka:

$$(n + 2) \times ((n+1) : 2)$$

Torej, če formula zares deluje, moramo dobiti formulo za $n + 1$ tudi tako, da formuli za n dodamo še vrednost $n + 1$, saj se v našem primeru vsota s povečanjem n - ja poveča za naslednje naravno število. Preizkusimo:

$$\begin{aligned} & (n + 1) \times (n : 2) + (n + 1) = \\ & = (n + 1) \times \frac{n}{2} + (n + 1) = \\ & = n \times \frac{n+1}{2} + (n + 1) = \\ & = n \times ((n + 1) : 2) + (n + 1) = \\ & = n \times ((n + 1) : 2) + 2 \times ((n + 1) : 2) = \\ & = ((n + 1) : 2) \times (2 + n) = \\ & = (n + 2) \times ((n+1) : 2) \end{aligned}$$

Po obračanju formule za n , ki ji prištejemo $n + 1$, dobimo kot rezultat formulo za $n + 1$! Tako smo dokazali, da če trditev velja za n , velja tudi za $n + 1$. Torej velja za vsa naravna števila!

$$1 \sim 2 \sim 3 \sim 4 \sim 5 \sim \dots$$

2. Primer - število podmnožic v množici

Dokazovanje začnemo na prvem ali prvih nekaj naravnih številih ($n = 1$). V našem primeru je to množica z enim elementom. Ker pravzaprav sploh še nimamo formule, ki bi jo dokazali, moramo med primerjanjem najti tudi to.

$$\text{Množica } 1 = \{1\}$$

Ugotovimo lahko, da ima množica z **1** elementom **2** podmnožici:

1. Prazno množico.
2. Samo sebe.

$$\text{Množica } 1, 2 = \{1, 2\}$$

Množica z **2** elementoma ima **4** podmnožice:

1. Prazno množico.
2. Množico 1. elementa.
3. Množico 2. elementa.
4. Samo sebe.

$$\text{Množica } 1, 2, 3 = \{1, 2, 3\}$$

Množica s **3** elementi ima **8** podmnožic:

1. Prazno množico.
2. Množico prvega elementa.
3. Množico 2. elementa.
4. Množico 3. elementa.

5. Množico 1. in 2. elementa.
6. Množico 2. in 3. elementa.
7. Množico 1. in 3. elementa in
8. samo sebe

...

Sedaj pogledajmo dobljena števila:

- 1 --- 2
- 2 --- 4
- 3 --- 8

Ugotovimo lahko, da so 2, 4, 8 potence števila 2. Torej smo pridobili domnevo:

Št. podmnožic = $2^{\text{št. elementov}}$.

Trditev 3: Množica z n elementi ima 2^n podmnožic.

Dokaz: Domneva je že dokazana na prvih nekaj številih, sedaj pa moramo dokazati *indukcijski korak*.

Če množici dodamo en element, se št. podmnožic z 2^n poveča na 2^{n+1} . To je isto, če bi 2^n pomnožili z 2, kar pomeni, da imamo z dodatkom 1 elementa 2 krat več podmnožic. Ugotovimo, da je tako zato, ker se vsem podmnožicam, ki jih množica že ima, dodajo prav enake, le da imajo dodan še novi element.

Množica = {1, 2, 3, 4}

- | | | |
|--------------|---|------------------|
| 1. { } | + | 9. {4} |
| 2. {1} | + | 10. {1, 4} |
| 3. {2} | + | 11. {2, 4} |
| 4. {3} | + | 12. {3, 4} |
| 5. {1, 2} | + | 13. {1, 2, 4} |
| 6. {2, 3} | + | 14. {2, 3, 4} |
| 7. {1, 3} | + | 15. {1, 3, 4} |
| 8. {1, 2, 3} | + | 16. {1, 2, 3, 4} |

Tako dokažemo, da če trditev velja za n , velja tudi za $n + 1$. Torej tudi ta trditev velja za vsa naravna števila.

Splošna pravila matematične indukcije

Z uporabo matematične indukcije na lažjih primerih smo končali. Oglejmo si še splošna pravila za uporabo le te.

1. Trditev dokažemo na primeru $n = 1$. Ta korak imenujemo *baza indukcije*.
2. Predpostavimo, da trditev velja za n in vsa predhodna števila.
3. Dokažemo induksijski korak ($n \rightarrow n + 1$).

Če trditev ustreza vsem tem pogojem, velja za vsa naravna števila.

Dokazovanje zmagovalne strategije

Zdaj, ko znamo uporabljati matematično indukcijo, lahko dokažemo tudi, ali naša zmagovalna strategija res drži. Dokazati hočemo, da je pri igri Nim zmagovalna strategija:

$$\begin{aligned} \text{porazdelitev pri } n \times 3 + 2 \text{ žetonih} &= [1, 0] \\ \text{porazdelitev pri } n \times 3 + 3 \text{ žetonih} &= [0, 1] \end{aligned}$$

Trditev 4: Optimalna strategija za igro Nim je, da, kadar imamo pred seboj $n \times 3 + 2$, vzamemo 1 žeton, kadar imamo pred seboj $n \times 3 + 3$ žetone, pa vzamemo 2 žetona. Kadar pred seboj vidimo $n \times 3 + 1$ žeton pa proti igralcu, ki igra s to strategijo, nimamo možnosti.

Dokaz: Pri tej strategiji je glavni namen spraviti soigralca do tega, da bo na začetku njegove poteze $n \times 3 + 1$ žetonov. S tem predpostavljamo, da je nasprotnik v teh situacijah poražen. Dokažimo to z matematično indukcijo.

Začnimo z $n = 0$. Ker je $0 \times 3 + 1 = 1$, je očitno, da smo poraženi, ko vidimo toliko žetonov, saj lahko vzamemo samo enega - zadnjega. To pomeni, da moramo pri $0 \times 3 + 2$ oziroma 2 žetonih vzeti 1 žeton, saj soigralcu tako ostane le še zadnji. Podobno je pri situaciji $0 \times 3 + 3$. Tokrat je rezultat izraza 3. Če želimo zmagati, moramo, kot tudi od nas zahteva strategija, vzeti 2 žetona. Tako našemu soigralcu zopet ostane en žeton.

Kaj pa, ko je $n = 1$? Formula za situacije, v katerih izgubimo, nam vrne število 4. Vemo že, da igralec, ki na začetku poteze vidi toliko žetonov proti nam, ki igramo z zmagovalno strategijo, nima možnosti. Številski izraz $1 \times 3 + 2$ ima rezultat 5. Če na začetku poteze vidimo 5 žetonov in nato vzamemo 1 žeton (kakor želi naša strategija), našemu soigralcu ostanejo 4. Kaj pa porazdelitev pri $1 \times 3 + 3$? Rezultat tega izraza je 6 in res - če bomo pri 6 žetonih vzeli 2, bodo soigralcu ostali 4.

Čas je za induksijski korak. Če na začetku poteze vidimo $(n + 1) \times 3 + 1$ žetonov, smo ne glede na to, ali vzamemo 1 ali 2 žetona, proti igralcu, ki igra z optimalno strategijo, nemočni. Če vzamemo 1 žeton, našemu soigralcu ostane $(n + 1) \times 3$ oziroma $n \times 3 + 3$. Ker za n trditev velja po predpostavki induksijskega koraka, nam poraz ne uide. Podobno se zgodi, če v situaciji $(n + 1) \times 3 + 1$ vzamemo dva žetona. Naš soigralec pred začetkom svoje poteze tako vidi $(n + 1) \times 3 - 1 = n \times 3 + 2$. Tudi ta primer nam je znan, saj po induksijski predpostavki našemu soigralcu ponuja možnost za zmago.

Zdaj pa še podrobneje dokažimo, da delujeta tudi drugi dve porazdelitvi. Če pred seboj vidimo $(n + 1) \times 3 + 2$ in upoštevamo strategijo ter vzamemo 1 žeton, našemu soigralcu ostane $(n + 1) \times 3 + 1$ žetonov. Kot smo že dokazali, v taki situaciji proti nam nima možnosti.

Dokazati moramo le še $(n + 1) \times 3 + 3$. Tukaj nam porazdelitev pravi, naj vzamemo dva žetona. Če to storimo, našemu soigralcu ostane $(n + 1) \times 3 + 1$. To število je spet za ena večje od večkratnika števila tri, zato tudi v tem primeru naš soigralec izgubi.

Iz tega je razvidno, da izgubimo, če vidimo $n \times 3 + 1$ žetonov, sicer pa lahko vedno pripravimo soigralca do te situacije, v kateri izgubi. Torej je naša zmagovalna strategija dokazana.

Računanje razdalje

V tem poglavju si pogledjmo, kako izračunamo razdaljo med enim in drugim elementom v koordinatnem sistemu. To znanje potrebujemo v eksperimentalnem delu, ko računamo, kolikšna je razdalja med optimalno strategijo in katerokoli drugo (oz. kako zelo se strategiji razlikujeta med seboj).

Razdalja na premici

Najprej si pogledjmo najenostavnejši primer - razdalja med dvema točkama na premici. Ta primer je najenostavnejši, ker je na vodoravni premici samo ena koordinata - x (levo / desno).

Recimo, da imamo premico s točko A, ki je na x koordinati 3, in točko B, ki je na koordinati 5.



Slika 10: Dve točki na premici.

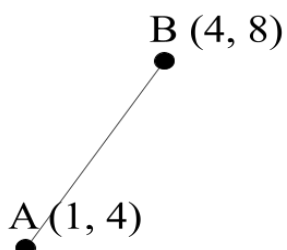
Če želimo izračunati njuno razdaljo, koordinato x od točke A preprosto odštejemo od koordinate x točke B.

$$\begin{aligned}d(A, B) &= \\&= 5 - 3 = \\&= 2\end{aligned}$$

Izračunamo, da je razdalja med točko A in točko B enaka 2.

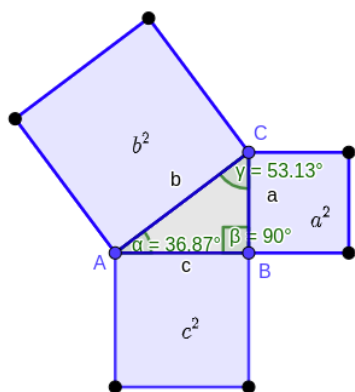
Razdalja v ravnini

V tem primeru spet iščemo razdaljo med dvema točkama. Toda tokrat sta prisotni dve koordinati - x (levo / desno) in y (gor / dol). Na **sliki 11** imamo točko A na koordinatah $x = 1$, $y = 4$ in točko B na koordinatah $x = 4$ in $y = 9$. Izračunajmo njuno razdaljo.



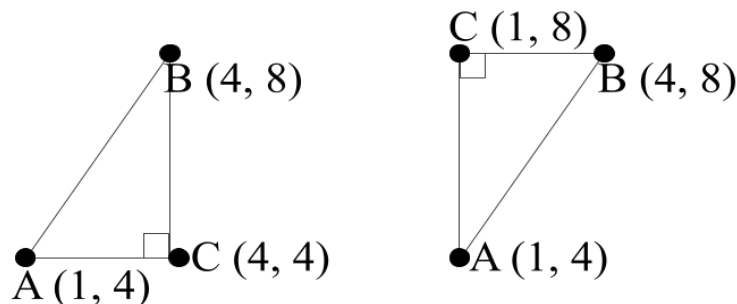
Slika 11: Dve točki v ravnini

Ker imamo dve koordinati, tokrat ne moremo zgolj odšteti, ampak moramo uporabiti *Pitagorov izrek*. Ta nam pove, da, če imamo pravokotni trikotnik, in z vsake njegove stranice naredimo kvadrat, je ploščina največjega kvadrata enaka vsoti ploščin manjših dveh. Tega pravila se poslužimo tako, da ustvarimo točko C, ki skupaj s prejšnjima točkama tvori pravokotni trikotnik. Nato izračunamo razdaljo med točkama A in C ter B in C (kar že znamo, saj je pomembna samo ena koordinata). Ko imamo dolžino teh stranic, lahko izračunamo njuno ploščino, ju seštejemo in dobimo ploščino tretjega kvadrata. S pomočjo le te pa lahko izračunamo razdaljo med točkama A in B, kar pa je tokrat naš cilj.



Slika 12: Pitagorov izrek.

Se pravi, najprej moramo določiti točko C. Če želimo, da je trikotnik pravokoten, mora imeti točka eno koordinato enako točki A, drugo pa enako točki B. Na voljo imamo dve možnosti, ki ju prikazuje **slika 13**.



Slika 13: Možnosti za pravokotni trikotnik

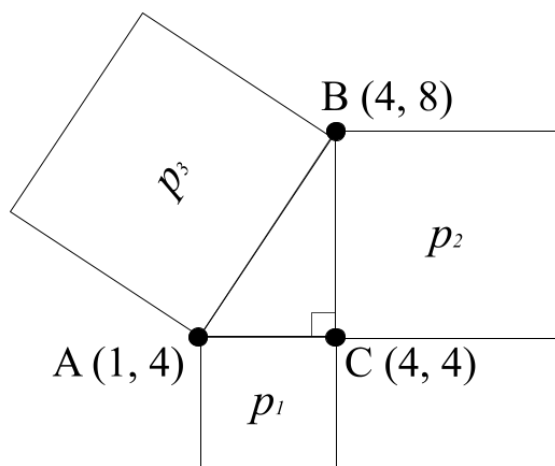
Ne glede na to, kateri primer izberemo, sedaj na enak način, kot v prvem primeru (razdalja na premici) izračunamo razdaljo med A in C ter med B in C.

Izberimo si npr. prvo sliko in izračunajmo razdalje:

$$d(A, C) = 4 - 1 = 3 \text{ (Upoštevamo samo koordinato x, saj imata y obe točki enak.)}$$

$$d(B, C) = 8 - 4 = 4 \text{ (Upoštevamo samo koordinato y, saj imata x obe točki enak.)}$$

Zdaj pa pride na vrsto Pitagorov izrek. Najbolje je, da ga kar grafično ponazorimo in izračunamo vse ploščine:



Slika 14: Tri ploščine

$$\begin{aligned} p_1 &= d(A, C)^2 = 3^2 = 9 \\ p_2 &= d(B, C)^2 = 4^2 = 16 \\ p_3 &= p_1 + p_2 = 9 + 16 = 25 \end{aligned}$$

Ugotovimo, da je ploščina tretjega kvadrata 25. Zdaj na podlagi tega podatka izračunamo, kako dolga je ena stranica tega kvadrata. Če želimo izvesti kaj takega, bomo potrebovali računsko operacijo, ki je nasprotna operaciji x^2 . To je *koren*, ki ga zapišemo tako: $\sqrt{x}$. Naša rešitev torej znaša $\sqrt{25} = 5$. Tako smo prišli do rešitve. Ker pa je takšen način računanja dokaj dolgotrajen, poiščimo v

literaturi (Bronštejn I. N., Semendjajev K. A. (1992)) formulo, ki naš postopek skrajša:

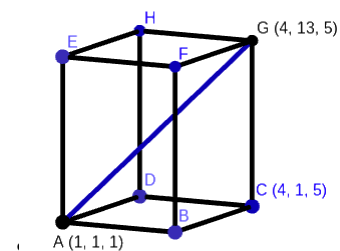
$$d(A, B) = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$$

Formula pove, da je razdalja med točkama A in B enaka korenu od vsote $(x_b - x_a)^2$ (to dobimo s pomočjo točke C in njene x koordinate) in $(y_b - y_a)^2$ (tudi to dobimo s pomočjo točke C in njene y koordinate).

Razdalja v prostoru

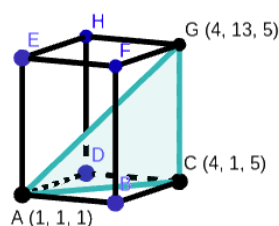
Zdaj smo prišli do situacije, ko točkam določimo 3 koordinate.

To so x (levo / desno), y (gor / dol) in z (naprej / nazaj). Za lažjo orientacijo iščemo kar razdaljo med dvema točkama (A, s koordinatami 1, 1, 1 in G, s koordinatami 4, 13, 5) v kvadru.



Slika 15: Dve točki v prostoru.

Rešimo problem tako, kot smo prej v dveh dimenzij - poiščimo točko, ki bo točki A in G dopolnjevala v pravokotni trikotnik. To je lahko v našem primeru ali točka C, s koordinatami 4, 1, 5 ali točka E, s koordinatami 1, 13, 1.

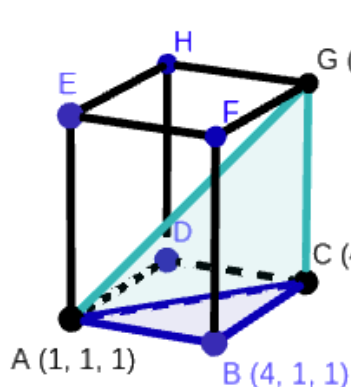


Ko naredimo daljici |AC| in |CG| dobimo zelen pravokotni trikotnik. Dolžino daljice |CG| lahko kar takoj izračunamo in pri tem upoštevamo le koordinato y:

$$\begin{aligned} d(C, G) &= \\ &= 13 - 1 = \\ &= 12 \end{aligned}$$

Slika 16: Pravokotni trikotnik v kvadru.

Za dolžino daljice |AC| pa potrebujemo še en pravokotni trikotnik, v katerega je lahko vključena ali točka B ali točka D. Ko si izberemo eno od njiju, še enkrat uporabimo Pitagorov izrek.



$$\begin{aligned} d(A, C) &= \\ &= \sqrt{d(A, B)^2 + d(B, C)^2} = \\ &= \sqrt{(4 - 1)^2 + (5 - 1)^2} = \\ &= \sqrt{3^2 + 4^2} = \\ &= \sqrt{9 + 16} = \\ &= \sqrt{25} = \\ &= 5 \end{aligned}$$

Slika 17: Dva pravokotna trikotnika v kvadru.

Tako dobimo še razdaljo med točkama A in C. To pomeni, da se lahko lotimo računanja daljice $|AG|$.

$$\begin{aligned} d(A, G) &= \\ &= \sqrt{d(A, C)^2 + d(C, G)^2} = \\ &= \sqrt{5^2 + 12^2} = \\ &= \sqrt{25 + 144} = \\ &= \sqrt{169} = \\ &= 13 \end{aligned}$$

Razdalja med točkama A in G je torej 13. Za konec pa poiščimo še splošno formulo za izračun razdalje dveh točk v prostoru. Ko imamo dve koordinati, je formula za razdaljo takšna: $d(A, B) = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$. Hitro lahko vidimo, da ta formula vsebuje dve koordinati. Formula za tri koordinate je podobna, treba ji je le dodati koordinato z. Zelo enostavno:

$$d(A, B) = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2 + (z_b - z_a)^2}$$

Splošna razdalja

Poglejmo si še, kako je z razdaljo med dvema točkama, kjer imamo n koordinat. Vemo že, da, ko dodamo novo koordinato, v bistvu le pod koren dodamo $+(x_b - x_a)^2$ (namesto x uporabimo simbol nove koordinate). Problem nastane, ker črk kmalu zmanjka. V takih primerih moramo uporabiti drugačen način zapisovanja koordinat. Lahko vse koordinate označimo z x in zaporedno številko:

$$d(A, B) = \sqrt{(x_{b,1} - x_{a,1})^2 + (x_{b,2} - x_{a,2})^2 + (x_{b,3} - x_{a,3})^2 + \dots + (x_{b,n} - x_{a,n})^2}$$

Računalniški igralci

Spoznavanje in dokazovanje strategij je za nami. Čas je, da se iskanja optimalne strategije loti še računalnik. Sprogramiramo ga tako, da se uči s poskušanjem, torej s *spodbujevanim učenjem*, spoznavanje katerega nas še čaka.

Na hitrost in kakovost učenja vpliva tudi izbor soigralca. Za to v programu ustvarimo nekaj simuliranih igralcev, ki igrajo z različnimi strategijami med seboj.

Živi igralec

Tega igralca upravlja človek.

Naključni igralec

Ko je na potezi naključni igralec, se za število vzetih žetonov odloči povsem naključno.

Naključni uteženi igralec

Naključni uteženi igralec se prav tako odloča naključno. Lahko pa mu damo razne uteži, s katerimi mu spremenimo porazdelitev za izbrane odločitve. Tako mu lahko mi sami določimo, kakšna bo njegova strategija.

Strateški igralec

Strateški igralec uporablja prej opisano zmagovalno strategijo. Ko mu ostane število žetonov, ki je za ena večje od večkratnika števila tri, pa vzame enega.

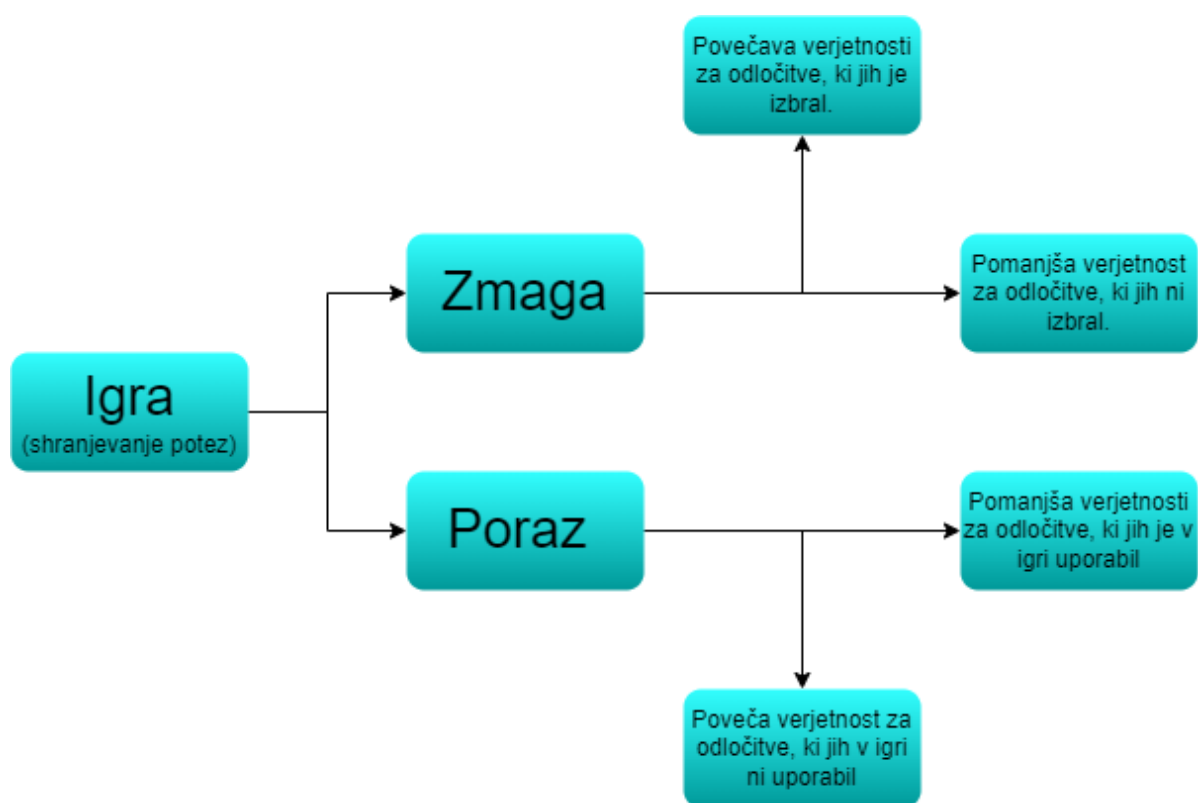
Strateški naključni igralec

Ta igralec uporablja prej opisano naključno zmagovalno strategijo. Če mu ostane število žetonov, ki je za ena večje od večkratnika števila tri (kar pomeni, da je na slabšem), pa se odloči naključno.

Učljivi igralec

Ta igralec strategijo pridobiva skozi odigrane igre. Torej se uči s spodbujevanim učenjem. Na Wikipediji (Wikipedija, Spodbujevano učenje) piše, da gre pri spodbujevanem učenju za strojno učenje, čigar cilj je priučiti ali optimizirati vedenje na podlagi povratne informacije (nagrajevanja oz. kaznovanja). "Spodbujevani učenec" v zanki izbira možnosti, ki so mu v dani situaciji na voljo. Če se izkaže, da je izbral pravilno možnost, je nagrajen, sicer pa kaznovan.

Pri spodbujevanem učenju se računalnik torej uči na podlagi odigranih iger. S simuliranim igralcem (enim od zgoraj opisanih) najprej večkrat opravi igro. Igrati začne z naključno strategijo - mešano strategijo, ki ima vse porazdelitve enake $[\frac{1}{2}, \frac{1}{2}]$. Po vsaki potezi si zabeleži, koliko žetonov vzame pri koliko žetonih. Če v igri zmaga, poveča verjetnost, da v prihodnjih igrah v teh situacijah spet vzame toliko žetonov, kot jih je vzel zdaj, in zmanjša verjetnost, da vzame število žetonov, ki jih v igri ni vzel. Če je v igri poražen, stori obratno.



Slika 18: Prikaz spodbujevanega učenja.

Za lažjo predstavo si oglejmo preprost primer 2 iger s 3 žetoni:
Recimo, da igro začne računalnik. Prva igra bi potem lahko izgledala tako:

Preglednica 2: Razlaga spodbujevanega učenja (prva igra).

Strategija pred igro	Igra	Rezultat igre	Strategija po igri
Porazdelitev pri 1 žetonu = $[1, 0]$ Porazdelitev pri 2 žetonih = $[\frac{1}{2}, \frac{1}{2}]$ Porazdelitev pri 3 žetonih = $[\frac{1}{2}, \frac{1}{2}]$	1. Računalnik vzame 1 žeton. 2. Soigralec vzame 1 žeton. 3. Računalnik vzame zadnji žeton.	Računalnik je poražen.	Porazdelitev pri 1 žetonu = $[1, 0]$ Porazdelitev pri 2 žetonih = $[\frac{1}{2}, \frac{1}{2}]$ Porazdelitev pri 3 žetonih = $[\frac{1}{4}, \frac{3}{4}]$

Računalnik po igri spozna, da ko ima pred seboj 3 žetone, ni vredno vzeti enega. Ker je to mogoče le naključje, še ne onemogoči tega, da bi spet kdaj vzel 1 žeton, ampak verjetnost za kaj takega vseeno zelo zmanjša. Do primera, ko ima pred seboj dva žetona, ne pride, zato porazdelitev ostaja enaka. Ker pa, ko ostane le še en žeton, ne more vzeti dveh, ostane porazdelitev tudi tu enaka.

Računalnik naslednjo igro začne igrati s strategijo, ki jo je razvil v prejšnji igri. Če je prva igra takšna, kot je v našem primeru, je druga najverjetneje taka:

Preglednica 3: Razlaga spodbujevanega učenja (druga igra).

Strategija pred igro	Igra	Rezultat igre	Strategija po igri
Porazdelitev pri 1 žetonu = $[1, 0]$ Porazdelitev pri 2 žetonih = $[\frac{1}{2}, \frac{1}{2}]$ Porazdelitev pri 3 žetonih = $[\frac{1}{4}, \frac{3}{4}]$	4. Računalnik vzame 2 žetona. 5. Soigralec vzame zadnji žeton.	Računalnik zmaga.	Porazdelitev pri 1 žetonu = $[1, 0]$ Porazdelitev pri 2 žetonih = $[\frac{1}{2}, \frac{1}{2}]$ Porazdelitev pri 3 žetonih = $[\frac{1}{8}, \frac{7}{8}]$

Računalnik je sedaj ustvaril strategijo, s katero pri 3 žetonih že zelo verjetno vzame 2 žetona. S ponavljanjem iger je verjetnost, da v takih okoliščinah vzame 1 žeton vse manjša.

Eksperimentalni del

Programiranje igralcev in igre

V tem razdelku je na kratko opisano, kako deluje naš program. Začnimo s prej opisanimi igralci. Vsak igralec je v programu predstavljen kot *razred*. Vsak razred ima svoje *spremenljivke* (v njih hranimo vrednosti, ki jih lahko spreminjamo) in *funkcije* (navodila, kaj naj razred naredi, ko je funkcija klicana). Funkcije imajo včasih *parametre*. To pomeni, da jim, ko jih kličemo, podamo nekaj podatkov. Ker so si vsi igralci zelo podobni, bomo opisali le tiste, ki nekoliko bolj izstopajo.

Razred Igralec

Ta igralec je najbolj osnoven. Vsebuje funkcije, ki so skupne vsem igralcem. Njegov razred v resnici ne počne ničesar. Njegov namen je, da lahko ostali igralci *podedujejo* njegove funkcije, in jih prilagodijo svojim potrebam. Funkcije, ki jih ima, so:

- `__init__()`: Ta funkcija se samodejno pokliče, ko je razred ustvarjen (še pred začetkom igre). Ponastavi spremenljivki `zmage` (ta šteje, kolikokrat je igralec zmagal) in `OpravljeneIgre` (v njej je shranjeno število iger, ki jih je igralec odigral).
- `__eq__(other)`: Tudi ta funkcija se samodejno pokliče takrat, ko primerjamo dva igralca, saj nas pogosto zanima, ali dva razreda pripadata isti vrsti igralca. Toda tu se pojavi težava, saj je vsak igralec svoja različica razreda. Tega zato ne moremo ugotoviti z navadnim primerjanjem (`a == b`). Ta funkcija zato spremeni način klasične primerjave, da lahko to vseeno izvedemo.
- `SemNaVrsti(NeVzeto)`: Izvede se, ko je igralec na potezi. Kot parameter dobi število žetonov, ki jih v igri še ni nihče odvezl - torej tistih, ki so trenutno na polju. Namen te funkcije je, da s pomočjo strategije, ki jo igralec uporablja, ugotovi, ali naj vzame 1 ali 2 žetona.
- `ObZmagi()`: Funkcijo `ObZmagi()` kličemo, ko igralec po koncu igre zmaga. Ko to storimo, funkcija poveča spremenljivki `OpravljeneIgre` in `Zmage`, ki, kot že njuni imeni povesta, štejeta, kolikokrat je igralec že zmagal in koliko vseh iger je odigral.
- `ObPorazu()`: Že iz imena lahko ugotovimo, da je ta funkcija klicana takrat, ko igralec izgubi. Tokrat se poveča le spremenljivka `OpravljeneIgre`.
- `IzpisiStrategijo()`: To funkcijo pokličemo, kadar želimo izvedeti, kakšno strategijo uporablja igralec. Funkcija vrne (programu, ki je funkcijo poklical, sporoči željeno vrednost / vsebino) strategijo, ki jo igralec uporablja.
- `JeRobot()`: Funkcija je klicana, kadar program potrebuje podatek, ali je igralec samoupravljaliv ali ga vodi človek. Ko je klicana, programu vrne `True`, če je igralec simuliran, oz. `False`, če ga upravlja človek.
- `SeUci()`: Kadar želimo vedeti, ali ima igralec točno določeno strategijo, ali pa le to razvija z učenjem, nas funkcija obvesti o tem.

- `PovejIme()` : Včasih nas zanima, katerega igralca opisuje neka spremenljivka tega tipa. Funkcija nam takrat vrne ime igralca (npr. Naključni igralec, Učljivi igralec itd.).
- `IzpišiGlavo()` : Vse, kar se v programu dogaja, se izpisuje v datoteko (tabelo), ki ima podatke ločene z vejicami (Comma Separated Values, .csv). Preden se pa izpisovanje začne, program pripravi prvo vrstico. Ampak ker pri različnih strategijah potrebujemo različno število stolpcev, program pokliče to funkcijo, da lahko vsak igralec po svojih potrebah prilagodi "glavo" tabele.
- `NarediNiz()` : Ko pokličemo to funkcijo, razred pripravi seznam podatkov, ki jih kasneje zapiše v datoteko. To so: ime igralca, število njegovih zmag in pri kolikšnem deležu iger je igralec zmagal.
- `PisiCSV(Datoteka)` : Ko je čas, da se v datoteko zapišejo podatki, funkcija v datoteko, ki jo igralec dobi kot parameter, izpiše podatke, ki jih igralec pripravi v funkciji `NarediNiz`.
- `NarediKlona(Nov)` : Funkcija je klicana, ko želimo ustvariti novega igralca. Če parameter zahteva, da je klon nov, program ustvari novega igralca istega tipa, kot je prejšnji. Če pa želi, da je igralec popolnoma enak prejšnjemu, pa stori enako, le da mu vse podatke (od števila zmag do strategije) nastavi tako, da imata oba enake.

Klasični igralci

Med klasične igralce uvrščamo *strateškega igralca* (ta igra z zmagovalno strategijo), *strateško naključnega* (v igri uporablja zmagovalno naključno strategijo) in *naključnega igralca* (ta uporablja naključno strategijo).

Uteženi naključni igralec

Klasični igralci igrajo s svojo strategijo, ta pa je na tem področju poseben. To, s kakšno strategijo igra, mu za vsako število žetonov, ki jih lahko vidi, določimo sami. Ker pa smo lahko površni in ne napišemo porazdelitve za vse situacije, ki jih igralec potrebuje, ima funkcijo, ki za vsako število žetonov, za katero mu nismo vnesli porazdelitve, ustvari porazdelitev $[\frac{1}{2}, \frac{1}{2}]$.

Učljivi igralec

Ta igralec prav tako nima točne strategije. Na začetku igra naključno, potem pa si jo nadgrajuje s spodbujevanim učenjem. Zato je nekaj njegovih funkcij prilagojenih. Tako npr. funkcija `SemNaVrsti()` ne pove samo, koliko žetonov bo igralec vzel, ampak si to tudi zabeleži, da bo lahko igralec na koncu igre popravil svojo strategijo. Za to ima tudi novo funkcijo - `Nauk()`, v kateri na prej opisan način dopolni svojo strategijo.

Igra in raziskava

Zdaj ko poznamo delovanje igralcev, si poglejmo še, kako so ti igralci uporabljeni. Vrnimo se k bistvu programa - pridobivanje podatkov o igrah. Očitno je, da če izvedemo več iger, dobimo točnejše podatke, zato morajo naši igralci izvesti kar se le da veliko iger. Pri tako množičnih podatkih pa hitro nastane zmeda, zato potrebujemo dober sistem. Mi uporabljamo naslednjega:

- Naš najbolj osnoven pojem je *igra*. V eni igri tekmujeta dva igralca. Eden od njiju zmaga, drugi pa je poražen.
- V eni *tekmi* se izvede veliko iger. Vse igre v eni tekmi odigrata dva **ista** igralca. To je pomembno predvsem takrat, ko je v tekmo vključen učljivi igralec. V eni sami igri namreč ne more ustvariti dobre strategije, zato vse igre v tekmi odigrata ista igralca.
- V enem *turnirju* se izvede več tekem. Če imamo turnir, v katerem želimo primerjati uspešnost npr. strateških igralcev z uspešnostjo npr. naključnih igralcev, program najprej izvede eno tekmo, kjer se pomerita en strateški in en naključni igralec. Ko se tekma konča, se začne druga tekma, v kateri se pomerita po en nov strateški in po en nov naključni igralec. Pri igralcih, ki imajo vedno enako strategijo, je povsem vseeno, če vse tekme odigra isti par ali drugačen par. Toda pri igralcih, kot je učljivi igralec, pa je to vsekakor potrebno, ker tako pri vsaki tekmi ustvari novo strategijo, in lahko dobimo veliko več podatkov o tem, pod kakšnimi pogoji igralec razvije boljšo strategijo.
- Naslednjo stopnjo raziskave imenujemo *primerjava uspešnosti*. V njej se izvede več turnirjev. Pri tem še vedno tekmujeta enaki skupini igralcev, le da se po vsakem novem turnirju spremeni število žetonov, ki so pred začetkom igre na polju. Tako lahko vidimo, kako prilagodljiva je kakšna skupina igralcev.
- Največji sklop pa predstavlja *raziskava*. Ta zagotovi, da se izvedejo vse možne kombinacije skupin igralcev.

Pri izvajanju tega programa računalnik zabeleži vse, kar se dogaja, in nato na podlagi tega ustvari diagrame, ki nam pregledno prikažejo najpomembnejše podatke.

Rezultati in razprava eksperimentalne raziskave

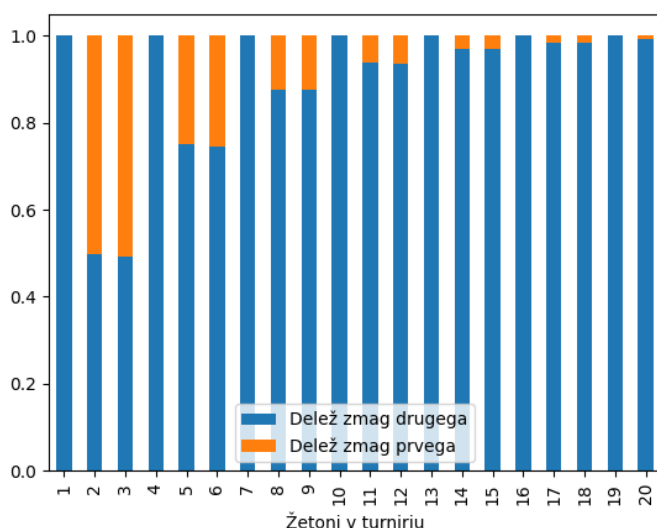
Med izvajanjem programa računalnik ustvari tri vrste grafov. V tem poglavju najprej predstavimo značilnosti vseh treh. V nadaljevanju jih uporabimo za prikaz rezultatov raziskave in razpravo o teh rezultatih.

Stolpični prikaz deleža zmag

x = žetoni v turnirju.

y = delež zmag prvega igralca (tistega, ki igro začne) in delež zmag drugega (tistega, ki pride na vrsto drugi).

Diagram nam prikaže razmerje med številom zmag prvega in drugega igralca za vsako število žetonov. **Slika 19** prikazuje primer, kjer je prvi igralec naključni (oranžno) in drugi igralec strateški (modro).

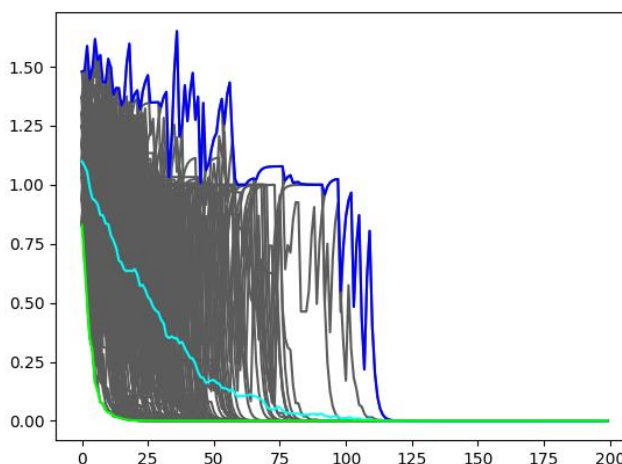


Slika 19: Diagram deleža zmag Naključni : Strateški.

Graf razdalje do optimalne strategije

x = zaporedno število igre

y = razdalja med strategijo, ki jo je pridobil učljivi igralec in optimalno strategijo. Kako razdaljo računamo, je opisano v teoretičnem delu. Pomembno je le, da tokrat pri računanju upoštevamo samo tiste primere, za katere ima optimalna strategija opredeljeno porazdelitev (torej ne upoštevamo števil žetonov $3 \times n + 1$).



Graf na **sliki 20** prikazuje razdaljo do optimalne strategije, kjer je na začetku igre 9 žetonov in je v igri poleg učljivega

Slika 20: Razdalja do optimalne strategije (Naključni, 9 žetonov).

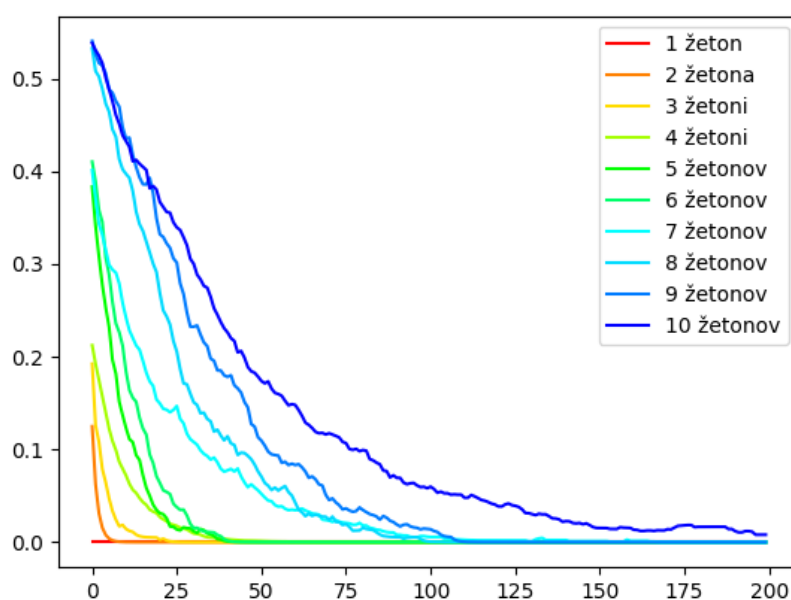
igralca še naključni. Vsaka siva krivulja predstavlja nihanje razdalje v eni tekmi. Temno modra krivulja označuje največjo razdaljo (kjer se igralec najpočasneje uči), svetlo modra krivulja predstavlja povprečje in zelena najmanjšo razdaljo (kjer se igralec najhitreje uči).

Graf povprečnega učenja

x = zaporedna igra

y = povprečna razdalja do optimalne strategije (za vsako število žetonov posebej).

Graf na **sliki 21** kaže povprečne razdalje do optimalne strategije pri vseh številih žetonov, kjer je v igri poleg učljivega še naključni igralec.



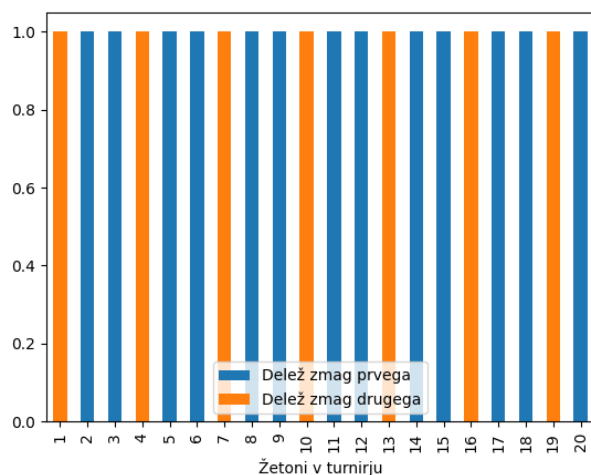
Slika 21: Povprečna razdalja do optimalne strategije (Naključni igralec).

V naslednjih poglavjih si pogledjmo rezultate vseh vrst grafov za različne pare igralcev in jih obrazložimo.

Diagrami deleža zmag

V prejšnjem poglavju je opisan videz tovrstnih prikazov. Poglejmo si dva takšna prikaza. To sta prikaz, kjer sta tako prvi kot tudi drugi igralec strateška, in prikaz, kjer je prvi igralec naključni, drugi pa strateški.

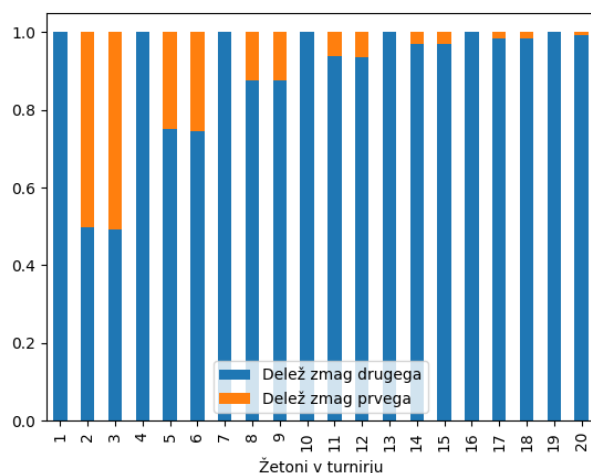
Strateški : strateški



Slika 22: Diagram deleža zmag Strateški : Strateški.

Vidimo lahko, da se je v vseh primerih zgodilo enako - ko je bilo na začetku igre $3 \times n + 1$ žetonov, je zmagal drugi igralec, v vseh preostalih primerih pa prvi igralec. To še dodatno potrdi pravilnost naše optimalne strategije.

Naključni : strateški



Slika 19: Diagram deleža zmag Naključni : Strateški.

O tem diagramu lahko povemo veliko več, saj se v njem pojavljajo zanimiva zaporedja. Tudi tokrat povsod, kjer je na začetku igre $3 \times n + 1$ žetonov, vedno zmaga drugi igralec (pozor - tokrat računalnik prvega igralca označi z oranžno, drugega pa z modro). Pri vseh ostalih

primerih pa nekajkrat zmaga tudi prvi (naključni) igralec, toda njegov delež zmag se z večjim številom žetonov zmanjšuje. Razmislimo, zakaj je tako.

Ko je na začetku igre en žeton, prvi igralec (naključni) vzame le tega in zmaga drugi (strateški).

Oglejmo si primere za $n = 0$.

Preglednica 4: Verjetnost za zmago naključnega proti strateškemu igralcu (prvi trije primeri).

Formula	Razlaga verjetnosti
$0 \times 3 + 1 = 1$	Ko naključni igralec vidi en žeton, za zmago nima možnosti. Verjetnost je torej 0.
$0 \times 3 + 2 = 2$	Kadar ima naključni pred sabo 2 žetona, je verjetnost, da vzame en žeton (in zmaga), enaka $\frac{1}{2}$, in prav tako verjetnost, da vzame dva žetona (in izgubi). Verjetnost za zmago naključnega igralca je zato $\frac{1}{2}$.
$0 \times 3 + 3 = 3$	Podobno kot prej – verjetnost, da vzame 1 žeton (in izgubi) je $\frac{1}{2}$, verjetnost, da vzame 2 žetona (in zmaga) je pa prav tako $\frac{1}{2}$.

Slika 23: Diagram deleža zmag
Naključni : Strateški (prvi trije primeri).

Podobno je tudi v primerih $n = 1$:

Preglednica 5: Verjetnost za zmago naključnega proti strateškemu igralcu (prvih šest primerov)

Formula	Razlaga verjetnosti
$1 \times 3 + 1 = 4$	Če naključni vidi 4 žetone, proti strateškemu nima možnosti.
$1 \times 3 + 2 = 5$	Kadar je na začetku poteze naključnega igralca 5 žetonov, je verjetnost, da ta vzame en žeton $\frac{1}{2}$ (če bi vzel dva, bi bil poražen). Če se odloči pravilno, ima lahko v naslednji potezi 2 ali 3 žetone. V obeh primerih je spet $\frac{1}{2}$ verjetnosti, da zmaga. Pri štirih žetonih je torej verjetnost, da zmaga $\frac{1}{2}$ od $\frac{1}{2}$, se pravi $\frac{1}{4}$.
$1 \times 3 + 3 = 6$	Primer je enak kot prej, le da mora za zmago najprej vzeti 2 žetona, ne 1.

Slika 24: Diagram deleža zmag
Naključni : Strateški
(prvih šest primerov).

Ta vzorec se ponavlja - kadar je na polju $n \times 3 + 1$ žeton, naključni ne more zmagati, v preostalih dveh primerih pa je verjetnost, da se odloči prav, enaka $\frac{1}{2}$. Takrat uspešno preide v naslednjo potezo, kjer je situacija podobna. Verjetnosti za njegovo zmago si sledijo nekako tako:

$$0, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{4}, \frac{1}{4}, 0, \frac{1}{8}, \frac{1}{8}, 0, \frac{1}{16}, \frac{1}{16}, 0 \dots$$

Tu lahko spet opazimo nekaj zanimivega - imenovalci so, tako kot v drugem primeru uporabe indukcije, potence števila dva. Najbolje, da jo še enkrat izvedemo, in dokažemo, da se verjetnosti tako nadaljujejo:

Trditev 5: Kadar naključni igralec igra proti strateškemu igralcu, v primerih $n \times 3 + 1$, za zmago nima možnosti. V primerih, ko vidi $n \times 3 + 2$ ali $n \times 3 + 3$ žetonov, pa je verjetnost za njegovo zmago enaka $\frac{1}{2^{n+1}}$.

Dokaz: Jasno je, da kadar igramo proti strateškemu igralcu, in je število žetonov na začetku igre $3 \times n + 1$, izgubimo. Dokaz za to najdemo v poglavju, kjer iščemo zmagovalno strategijo.

Baza indukcije za $n = 0$ in $n = 1$ je obravnavana pred trditvijo. Preostane nam torej le še indukcijski korak za dokazovanje verjetnosti pri ostalih številih žetonov.

Kot že vemo, gre pri indukcijskem koraku za to, da predpostavimo, da trditev velja za vsa naravna števila do n (vključno z njim), in na podlagi predpostavljenega dokažemo, da trditev velja tudi za $n + 1$, s tem pa vsa naravna števila.

Lotimo se dela tako, da trditev za n spremenimo v trditev za $n + 1$. To bi bilo nekako tako:

Kadar naključni igralec igra proti strateškemu igralcu pri situacijah $(n + 1) \times 3 + 1$ za zmago nima možnosti. V primerih, ko vidi $(n + 1) \times 3 + 2$ ali $(n + 1) \times 3 + 3$, pa je verjetnost za njegovo zmago enaka $\frac{1}{2^{n+2}}$.

Dokažimo zdaj to trditev tako kot pri prvih nekaj številih. Kadar naključni igralec pred seboj vidi $(n + 1) \times 3 + 2$, je verjetnost, da vzame dva žetona, enaka $\frac{1}{2}$. Če se zgodi ta dogodek, njegov soigralec (strateški) pred seboj vidi $(n + 1) \times 3$ žetonov, kar je enako kot $n \times 3 + 3$. Ker indukcijski korak predpostavlja, da za n trditev velja, vemo, da lahko zdaj strateški igralec zmaga.

Verjetnost, da naključni vzame 1 žeton, je prav tako $\frac{1}{2}$. Tako strateškemu ostane $(n + 1) \times 3 + 1$. V tem primeru pa že vemo, da je na slabšem. Če strateški nato vzame 1 žeton, naključni po tem vidi $n \times 3 + 3$. Če pa vzame 2 žetona, naključni svojo potezo začne z $n \times 3 + 2$. Ker smo predpostavili, da za n trditev velja, je verjetnost, da naključni zmaga, enaka $\frac{1}{2^{n+1}}$. Verjetnost, da naključni igralec zmaga v situaciji $(n + 1) \times 3 + 2$, je enaka $\frac{1}{2} \times \frac{1}{2^{n+1}} = \frac{1}{2^{n+1} \times 2} = \frac{1}{2^{n+2}}$.

Iz podobnih razlogov velja trditev tudi v situaciji $(n + 1) \times 3 + 3$. Verjetnost, da naključni igralec vzame en žeton, je enaka $\frac{1}{2}$. Če se to zgodi, strateškemu ostane $(n + 1) \times 3 + 2$

žetonov. V takem primeru že vemo, da strateški igralec zmaga. Je pa tudi verjetnost, da naključni vzame 2 žetona, enaka $\frac{1}{2}$. Tako strateškemu ostane $(n + 1) \times 3 + 1$. Točno ta primer najdemo tudi v prejšnjem odstavku - verjetnost, da naključni zmaga, je torej tudi tu enaka $\frac{1}{2^{n+2}}$. Trditev je tako dokazana.

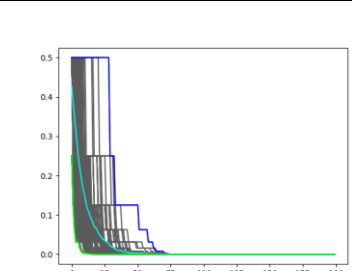
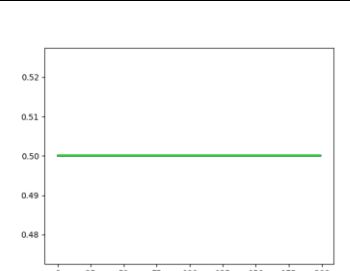
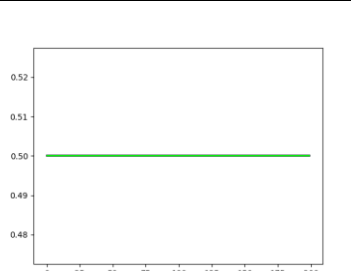
Graf razdalje do optimalne strategije

V tem poglavju se spet poglobimo v našega učljivega igralca in ugotovimo, kako izbor soigralca vpliva na hitrost učenja.

$$n \times 3 + 1$$

Začnimo s primeri, kjer se igra začne s štirimi žetoni ($n \times 3 + 1$).

Preglednica 6: Razdalja do optimalne strategije pri $n \times 3 + 1$.

Naključni igralec	Strateški naključni igralec	Strateški igralec
 Slika 25: Razdalja do optimalne strategije (Naključni, 4 žetoni)	 Slika 26: Razdalja do optimalne strategije (Strateški naključni, 4 žetoni)	 Slika 27: Razdalja do optimalne strategije (Strateški, 4 žetoni)

Učljivi igralec se v tekmah z naključnim igralcem nekaj nauči, saj lahko naključnega premaga. V igri s strateškim naključnim in strateškim igralcem pa se žal ne more ničesar naučiti, saj je v tisti "nemogoči situaciji", kjer ga lahko ta dva igralca brez težav premagata. Ugotovimo, da je v nekaterih primerih za učenje primernejši naključni igralec.

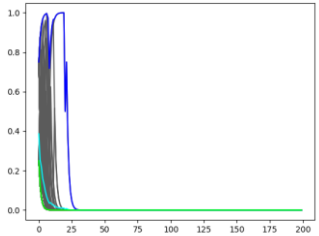
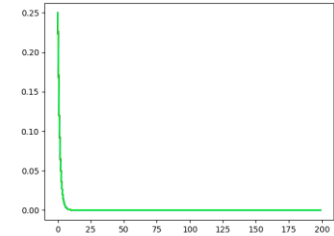
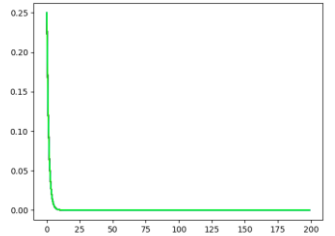
$$n \times 3 + 2, n \times 3 + 3$$

Zdaj, ko je situacija $n \times 3 + 1$ obdelana, se lahko lotimo še preostalih, učenju prijaznejših primerov. Ker lahko sklepamo, da več, kot je žetonov, težje se je učiti, si oglejmo rezultate v dveh skupinah:

Manjša števila

V **preglednici 7** so prikazani grafi učenja pri treh žetonih.

Preglednica 7: Razdalja do optimalne strategije pri nižjih številih.

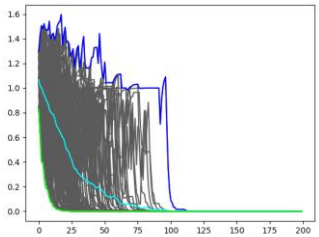
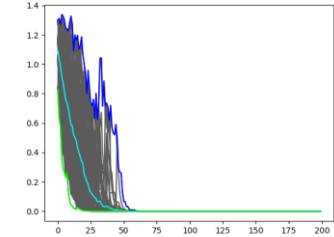
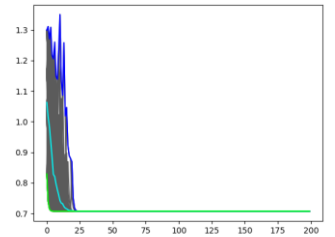
Naključni igralec	Strateški naključni igralec	Strateški igralec
 Slika 28: Razdalja do optimalne strategije (Naključni, 3 žetoni).	 Slika 29: Razdalja do optimalne strategije (Strateški naključni, 3 žetoni).	 Slika 30: Razdalja do optimalne strategije (Strateški, 3 žetoni).

Tokrat se igralec nauči strategije v vseh primerih dokaj hitro (pri naključnem igralcu seveda malo počasneje).

Večja števila

Preglednica 8 nam kaže učne krivulje pri osmih žetonih.

Preglednica 8: Razdalja do optimalne strategije pri višjih številih.

Naključni igralec	Strateški naključni igralec	Strateški igralec
 Slika 20: Razdalja do optimalne strategije (Naključni, 9 žetonov).	 Slika 31: Razdalja do optimalne strategije (Strateški naključni, 9 žetonov).	 Slika 32: Razdalja do optimalne strategije (Strateški, 9 žetonov).

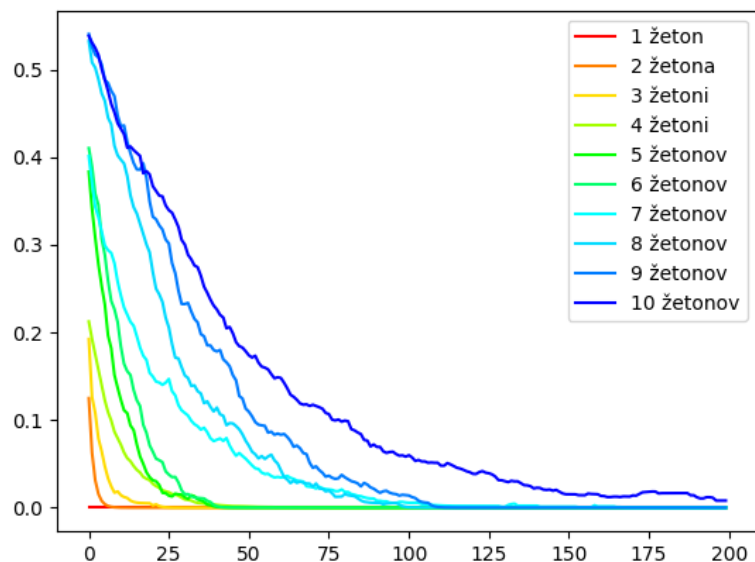
Vidimo, da se tokrat učljivi igralec uči nekoliko počasneje, sicer pa je stvar podobna.

Povprečno učenje

Sledi nam le še ogled grafov povprečnega učenja. Tokrat imamo za vsakega računalnikovega soigralca svoj graf.

Učljivi in naključni igralec

Ko računalnik igra z naključnim igralcem, dobimo graf, ki ga prikazuje **slika 21**:

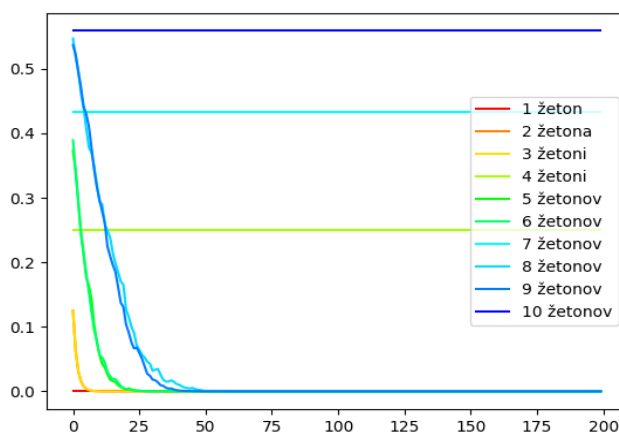


Slika 21: Povprečna razdalja do optimalne strategije (Naključni igralec).

Vidimo, da manj, kot je žetonov, prej igralec pridobi optimalno strategijo. Ker se naključni igralec vedno odloča naključno, lahko učljivi zmaga, tudi če stori napako. V takih primerih je nagrajen, kljub temu da se ni ravnal, tako kot bi moral. Zaradi tega potrebuje veliko časa, da pride do želene strategije.

Učljivi in strateški naključni igralec

Na **sliki 33** vidimo, da se učljivi igralec pri številu žetonov, ki ni $3 \times n + 1$, dokaj hitro nauči optimalno igrati, predvsem pa bistveno hitreje od takrat, ko igra proti naključnemu igralcu. To se kaže tudi v realnosti - če nas uči učitelj, ki je vešč učenja, se hitro naučimo, če pa nas uči manj vešč učitelj, se učimo.



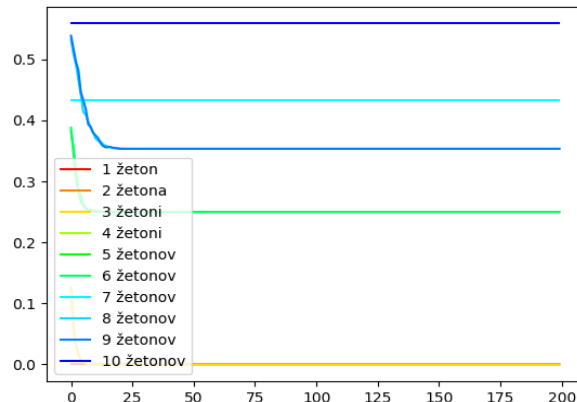
Slika 33: Povprečna razdalja do optimalne strategije (Strateški naključni igralec).

Toda kljub počasnejšemu učenju ima računalnik pri učenju z naključnim igralcem nekaj prednosti. Do optimalne strategije namreč pride v vseh primerih, saj se učljivi igralec lahko

nauči izkoristiti napako naključnega, proti strateškemu naključnemu pa v primerih $3 \times n + 1$ nima možnosti in se ničesar ne nauči.

Učljivi in strateški igralec

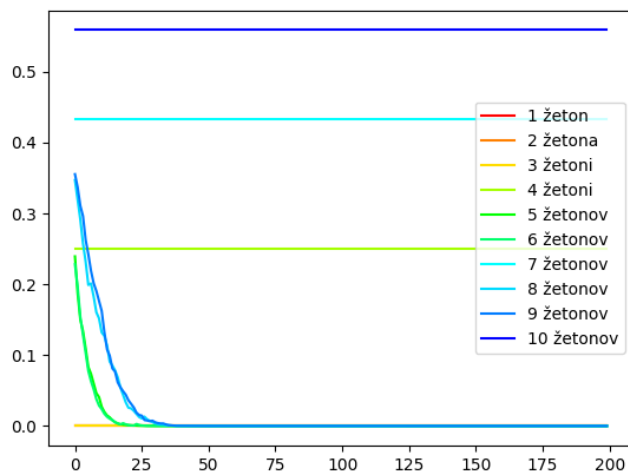
Človek bi pričakoval, da je z rezultati pri strateškem igralcu dokaj podobno, če ne še bolj. Toda na **sliki 34** se izkaže, da ni tako. Zakaj le?



Slika 34: Povprečna razdalja do optimalne strategije (Strateški igralec).

Ker ta igralec, ko ima pred sabo $3 \times n + 1$ žetonov, vedno vzame enega, se učljivi igralec nikoli ne sreča z nekaterimi situacijami. Pri njih tako porazdelitev ostane $[0.5, 0.5]$ in zato sicer razvije strategijo, s katero premaga strateškega igralca, toda razdalja med njo in optimalno strategijo ni nikoli 0. Če tudi to primerjamo z realnostjo, lahko rečemo, da mora učitelj dopustiti, da spoznamo vse zanimive situacije. Če se z določenimi nikoli ne srečamo, se jih brez razmisleka ne moremo naučiti obvladati.

Da pa učljivi igralci svojega dela ne pustijo nedokončanega, smo jih sprogramirali tako, da ko končajo s tekmami proti strateškemu igralcu, si shranijo svojo strategijo in z njo začnejo še boj proti strateškim naključnim igralcem. Graf nadaljevanja učenja prikaže, da se strategijo naučijo do konca, saj se v nadaljevanju srečajo z vsemi možnimi primeri.



Slika 34: Povprečna razdalja do optimalne strategije (nadaljevano učenje s strateškim naključnim igralcem).

Zaključki

V raziskovalni nalogi *Igra Nim skozi matematiko in spodbujevano učenje* raziskujemo iskanje strategij z matematiko, z računalnikom in razliko med tema dvema.

Pri matematičnem iskanju strategije v teoretičnem delu z opazovanjem igre od zadnje proti prvi potezi pridobimo vse potrebne informacije o optimalni strategiji. Matematična indukcija nam omogoča, da dokažemo, da naša strategija velja za vsa naravna števila žetonov. Tako lahko človek na podlagi le nekaj primerov pridobi strategijo, ki bo veljala čisto povsod. Temu rečemo, da zna znanje *posplošiti*.

V eksperimentalnem delu pokažemo, da zna do strategije priti tudi računalnik in sicer s spodbujevanim učenjem. Toda on mora za to, da pridobi optimalno strategijo, izvesti res veliko iger. In ne samo to, strategija, ki jo pridobi, je uporabna le za toliko žetonov, kot jih je na začetku v njegovi igri ali manj, ne pa za več. Ni mogoče, da bi se tako naučil strategije za n žetonov, saj svojega znanja ne zna posplošiti.

Računalnikova strategija in hitrost, s katero jo je pridobil, je zelo odvisna od tega, s katerim simuliranim soigralcem igra igro. Kadar igra proti naključnemu igralcu, strategijo dobi v vseh primerih. Ker pa je učljivi igralec včasih zmaga kljub temu, da je storil napako, se uči dokaj počasi.

Ko se v igri pomeri s strateškim igralcem, se pri situaciji, kjer se igra začne z $n \times 3 + 1$ žetoni, nič ne nauči - ker je vedno poražen. V situacijah, kjer se igra začne s številom žetonov, pri katerih lahko zmaga, pa pridobi strategijo, s katero sicer premaga nasprotnika, toda le ta ni optimalna, saj zaradi lastnosti strateškega igralca (pri $n \times 3 + 1$ žetonih vedno vzame enega) ne spozna vseh situacij in se jih ne nauči obvladati.

Kadar pa igra proti strateškemu naključnemu igralcu, optimalno strategijo hitro pridobi v vseh primerih, ko se igra začne z $n \times 3 + 2$ ali $n \times 3 + 3$ žetonov in to čisto do konca, pri $n \times 3 + 1$ žetonih pa ne.

Ugotovimo, da se računalnik hitreje uči, kadar igra proti igralcu, ki igra z optimalno strategijo. Žal pa to ne velja povsod - v primeru $n \times 3 + 1$ se namreč strategije nauči le v igri z naključnim igralcem. S tem poleg vsega pokažemo tudi splošno resnico, da se, kadar nas uči učitelj, ki je večš učenja, hitro naučimo, če pa je učitelj manj večš, se naučimo počasneje (je pa to odvisno tudi od nas učencev).

Viri in literatura

Bronštejn J. N., Semendjajev K. A. (1992) Matematični priročnik. Tehniška založba Slovenije, Ljubljana.

Jamnik R. (1973) Teorija iger. Državna založba Slovenije, Ljubljana.

Kavka D., Pavlič G., Rugelj M., Šparovec J. (2016) Tempus novum. Modrijan, Ljubljana.

Krebelj P. (2016) Spoznavamo programski jezik python. Doria, Ljubljana.

Teja Rajgelj, spletni vir, http://www.educa.fmf.uni-lj.si/izodel/sola/2002/dira/tea/uvod_v_verjetnost.html, prevzeto 5. 9. 2021.

Spodbujevano učenje, spletni vir, https://sl.wikipedia.org/wiki/Spodbujevano_u%C4%8Denje, prevzeto 21. 1. 2022.