

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Mateja Rojko

# **Ideje in gradiva v računskem razmišljanju**

DIPLOMSKO DELO  
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN MATEMATIKA

Mentor: prof. dr. Marko Robnik Šikonja

Ljubljana, 2012



Št. naloge: 00004/2011

Datum: 05.12.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko ter Fakulteta za matematiko in fiziko izdaja naslednjo nalogo:

Kandidat: **MATEJA ROJKO**

Naslov: **IDEJE IN GRADIVA V RAČUNSKEM RAZMIŠLJANJU**  
**IDEAS AND RESOURCES IN COMPUTATIONAL THINKING**

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

Računsko razmišljanje je način reševanja in razumevanja problemov, ki posplošuje metode in tehnike, ki se uporabljajo v računalništvu, tudi na druga področja. Gre tako za splošne kot za specializirane pristope iz načrtovanja algoritmov, na primer: razčlenitev na podprobleme, posplošitev in abstrakcija problemov, razpoznavanje vzorcev, analiza, organizacija in modeliranje podatkov, itd. V svetu se računskemu razmišljanju posveča vse več pozornosti in se ga poskuša čim bolj zgodaj vpeljati v izobraževalni proces, ne le v obliki računalniškega programiranja, pač pa v obliki nalog in primerov, ki naj učence naučijo splošnih pristopov reševanja problemov.

Proučite primere dobre prakse in zanimivih prenosov računskega razmišljanja v poučevanje računalništva ter drugih področij. Izdelajte pregled zanimivih idej in koristnih gradiv za učitelje računalništva. Izdelajte lasten učni primer računskega razmišljanja, idejo ustrezno opišite ter predstavite na zanimivem problemu.

Mentor:

  
prof. dr. Marko Robnik Šikonja



Dekan Fakultete za računalništvo in informatiko:

  
prof. dr. Nikolaj Zimic

Dekan Fakultete za matematiko in fiziko:

akad. prof. dr. Franc Forstnarič





# **IZJAVA O AVTORSTVU diplomskega dela**

Spodaj podpisana **Mateja Rojko,**

z vpisno številko **63050212,**

sem avtorica diplomskega dela z naslovom:

**Ideje in gradiva v računskem razmišljanju**

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom **prof. dr. Marka Robnika Šikonje**
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 13.3.2012

Podpis avtorice:

# Zahvala

V prvi vrsti se zahvaljujem prof. dr. Marku Robniku Šikonji za pomoč in usmerjanje pri izdelavi diplomske naloge.

Zahvala gre tudi moji družini in prijateljem za vso potrpežljivost in spodbude v času študija.

# Kazalo

<b>Povzetek</b>	<b>1</b>
<b>Abstract</b>	<b>3</b>
<b>1 Uvod</b>	<b>5</b>
<b>2 Opis računskega razmišljanja</b>	<b>9</b>
2.1 Začetki.....	9
2.2 Tehnike in metode v računskem razmišljanju.....	10
2.2.1 Dekompozicija .....	10
2.2.2 Razpoznavanje vzorcev.....	11
2.2.3 Abstrakcija .....	12
2.2.4 Načrtovanje algoritmov.....	13
2.3 Računsko razmišljanje v vsakdanjem življenju .....	14
2.4 Računsko razmišljanje v drugih disciplinah .....	15
2.4.1 Kemija .....	15
2.4.2 Biologija.....	16
2.4.3 Novinarstvo .....	17
2.4.4 Igre .....	17
2.4.5 Univerzitetno izobraževanje.....	18
2.4.6 Izobraževanje v K - 12 .....	18
2.4.7 Ostala področja.....	20
<b>3 Konkretni primeri iz posameznih področij</b>	<b>23</b>
3.1 Robotska kirurgija.....	23
3.2 Shotgun algoritem .....	25
3.3 Optimalna izmenjava ledvic.....	29
3.4 Igra RoboZZle.....	30
3.4.1 RoboZZle ukazi.....	30
3.4.2 Rekurzivne funkcije .....	31

3.5 Kaj je algoritem in kaj algoritmično razmišljanje za otroke .....	34
<b>4 Urejanje z zlivanjem</b>	<b>37</b>
4.1 Opis algoritma .....	37
4.2 Lastnosti algoritma .....	40
4.2 Animacija urejanja Lego kock.....	41
4.3.1 Orodje Adobe Flash.....	41
4.3.2 Animacija v slikah .....	41
<b>5 Zaključek</b>	<b>49</b>
<b>Seznam slik</b>	<b>51</b>
<b>Seznam tabel</b>	<b>53</b>
<b>Literatura</b>	<b>55</b>

# Seznam uporabljenih kratic in simbolov

**PROBE** – problemsko usmerjene raziskave (angl. Problem-oriented Exploration)

**ACM** – največja svetovna izobraževalna in računalniška družba, ki zagotavlja sredstva za napredek v računalništvu (angl. Association for Computing Machinery)

**K - 12** – osnovna stopnja izobraževanja v ZDA. K – 12 vključuje otroke od vrtca do 12. razreda (16 – 19 let starosti)

**NSF** – nacionalna ustanova za znanost je državna vladna agencija v ZDA, ki podpira temeljne raziskave in izobraževanje na vseh nezdravstvenih področjih znanosti in tehnike (angl. The National Science Foundation)

**ISTE** – mednarodno združenje za tehnologijo v izobraževanju (angl. The International Society for Technology in Education)

**CSTA** – društvo učiteljev računalništva (angl. The Computer Science Teachers Association)

**FDA** – ameriški vladni urad za zdravila in prehrano, ki je pod okriljem ameriškega ministrstva za zdravje (angl. Food and Drug Administration)

**LIFO** – kratica, ki pomeni zadnji noter, prvi ven. V računalništvu se uporablja pri shranjevanju podatkov za kratek čas. Tisti podatek, ki je zadnji prišel v neko vrsto, jo bo prvi zapustil (angl. Last In, First Out)

# Povzetek

Računsko razmišljanje je način reševanja problemov, načrtovanja sistemov in razumevanja človeškega obnašanja, ki črpa iz temeljnih konceptov računalništva. Temelj računskega razmišljanja je abstrakcija – abstrahiranje pojmov iz primerov in ocenjevanje ter izbiranje ustrezne abstrakcije. Vpliv računskega razmišljanja je razširil na raziskovanje v znanstvenih kot tudi humanističnih vedah. V izobraževanju se vključuje na področje matematike, biologije, kemije, ekonomije, financ, jezikoslovja in tudi umetnosti, glasbe in športa. Otroke lahko začnemo že zelo zgodaj s pomočjo iger seznanjati z osnovnimi koncepti računalništva in računskega razmišljanja. V diplomskem delu predstavimo metode, ki se pojavljajo pri računskem razmišljanju, kot so dekompozicija, abstrakcija in načrtovanje algoritmov. Opišemo področja, na katera vpliva računsko razmišljanje, pri čemer je večji poudarek na vključevanju računskega razmišljanja v kurikulum osnovnih in srednjih šol. Podrobneje opišemo nekaj konkretnih primerov iz posameznih področij, kot je npr. robotska kirurgija v medicini in »shotgun« algoritem s področja biologije. S programom Flash smo ustvarili animacijo, s pomočjo katere lahko na zanimiv in preprost način otroke naučimo urejanja z zlivanjem.

**Ključne besede:**

računsko razmišljanje, abstrakcija, dekompozicija, računalniški koncepti, urejanje z zlivanjem, reševanje problemov, algoritmično razmišljanje





# Abstract

Computational thinking is a paradigm of solving problems, designing systems and understanding human behavior by drawing on the fundamental concepts of computer science. The foundation for computational thinking is abstraction - abstracting concepts from cases and evaluating and selecting the appropriate abstraction. Computational thinking has already influenced a research in many science and engineering disciplines. In education, it is part of mathematics, biology, chemistry, economics, finance, languages and arts, music and sport. Through the games we can teach children the basic concepts of computer science and computational thinking. We present characteristics and methods of computational thinking, such as decomposition, abstraction and design of algorithms. We describe areas affected by the computational thinking, but we focus on their involvement in K – 12 curriculum. We describe some examples, such as robotic surgery in the medicine and »shotgun« algorithm in biology. We create a Flash animation, which tends to teach children merge sort in an interesting and simple manner.

**Key words:**

computational thinking, abstraction, decomposition, computer concepts, merge sort, problem solving, algorithmic thinking



# 1 Uvod

Ne dolgo nazaj si je bilo težko predstavljati, da bi lahko vsak od nas v rokah držal mobilno napravo, ki bi imela hitrost in moč namiznega računalnika. V prvih dneh statičnih spletnih strani so si le redki predstavljali potencial svetovnega spleta. V primerjavi z danes, je takrat splet spominjal na brošuro. Danes je to prostor za informiranje in iskanje virov, komuniciranje in objavljane, sodelovanje in učenje, ustvarjanje ter druženje. Te in še mnoge druge inovacije je omogočila računalniška tehnologija, ki predstavlja rezultat del in ustvarjanja številnih računalničarjev, računalniški inženirji in oblikovalci.

Ogromen napredek na področju računalništva je začel spreminjati način vsakdanjega življenja. Računalnik lahko v igranju šaha premaga najboljšega šahista na svetu. Lahko prepozna obraz in nas usmerja, ko se hočemo pripeljati na določen kraj. Veliki skoki v inovativnosti in domišljiji so privedli do dekodiranja človeškega genoma, prevodov antičnih jezikov in sistemov za sledenje. Celotno zbirko glasbe lahko imamo v žepu. Otroci za zabavo ustvarjajo lastne aplikacije in spletne igre. Podjetniki ustvarjajo nove tehnologije in storitve. Računalniški potencial za inovacije in ustvarjalnost novih modelov je neomejen.

Ena izmed zanimivejših stvari pri učenju računalništva je, da se naučimo novih in temeljnih načinov razmišljanja in reševanja problemov. Razmišljanje, ki je ključnega pomena v 21. stoletju, je tako imenovano računsko razmišljanje. Tako kot matematično razmišljanje črpa ideje iz matematike in statistično razmišljanje iz statistike, temelji računsko razmišljanje na osnovah računalništva. Računsko razmišljanje uči, kako uporabljati abstrakcijo in dekompozicijo pri reševanju zapletenih primerov. Predstavlja okvir za razumevanje algoritmov, opisuje bistvene koncepte pri ravnanju s podatki in pri izražanju omejitev sodobne računalniške opreme. Računsko razmišljanje temelji na moči in omejitvah računskih procesov, ne glede na to ali jih izvajajo ljudje ali stroji. Vključuje reševanje problemov, načrtovanje sistemov in razumevanje obnašanja ljudi, pri tem pa črpa iz temeljnih konceptov računalništva.

Danes so računalniki in računsko razmišljanje ključni del mnogih disciplin. Pri pohitritvi sekvenciranja človeškega genoma je bil uporabljen algoritem »shotgun«. V ekonomiji avtomatizirani mehanizmi oblikujejo temelje elektronskega poslovanja, kot

sta spletno oglaševanje in spletne dražbe. Robotske operacije, virtualne kolonoskopije in elektronske zdravstvene kartoteke so primer računskega razmišljanja v medicini. S pomočjo k-d dreves astronomi analizirajo velike večdimenzionalne zbirke podatkov. Računalniška grafika je močno spremenila filmsko industrijo. Računsko razmišljanje ima vpliv tudi v disciplinah kot so glasba, umetnost in fotografija. Največjo vlogo pa igra računsko razmišljanje na področju izobraževanja. Mnogi si prizadevajo za vključitev računskega razmišljanja v kurikulum K-12. Na spletnih straneh lahko najdemo veliko že pripravljenih gradiv za poučevanje otrok. Univerze ustanavljajo nove programe, ki povezujejo računsko razmišljanje z drugimi disciplinami.

Računsko razmišljanje je pri nas še relativno neznanu področje. Zato je namen diplomske naloge predstaviti to področje, naštetih ter opisati osnovne lastnosti tega koncepta in metode, ki jih vključuje, prav tako pa izpostaviti pomembnost vključevanja računskega razmišljanja v različne discipline, še posebej na področju osnovnošolskega in srednješolskega izobraževanja. S tem namenom smo ustvarili animacijo, ki prikazuje algoritem urejanje z zlivanjem, na način ki je razumljiv in blizu otrokom.

V prvem delu diplomske naloge si bomo pogledali začetke računskega razmišljanja. Predstavili bomo tudi inštitut, ki je center računskega razmišljanja. V nadaljevanju bomo predstavili in opisali tehnike in metode, ki so najpogostejše pri računskem razmišljanju, kot so dekompozicija, abstrakcija, razpoznavanje vzorcev in načrtovanje algoritmov. Pri vsaki od teh metod bomo navedli tudi konkreten primer. Za zaključek prvega dela bomo predstavili vlogo računskega razmišljanja v drugih disciplinah in v vsakdanjem življenju.

V drugem delu bomo naštetih in opisali nekaj primerov iz posameznih področij. Predstavili bomo algoritem »shotgun«, ki je pripomogel k razvozanju človeškega genoma. Opisali bomo tudi program izmenjave ledvic, ki postaja vse bolj razširjen po celem svetu. Otroci se lahko naučijo osnovnih računalniških konceptov tudi preko številnih miselnih iger. Zato bomo v tem poglavju opisali igro RoboZZle, s pomočjo katere se lahko naučimo osnov programiranja z uporabo funkcij in rekurzije. Kot zadnji primer pa si bomo pogledali eno izmed lekcij, ki so objavljene na strani, ki jo je Google razvil za pomoč učiteljem pri vključevanju računskega razmišljanja v pouk. S to lekcijo se učenci naučijo, kaj je algoritem in algoritmično razmišljanje.

V tretjem delu bomo opisali algoritem urejanja z zlivanjem, naštetih njegove časovne in prostorske karakteristike, ter ga primerjali z algoritmom Quicksort. S pomočjo animacije izdelane v programu Flash bomo predstavili urejanje z zlivanjem.

Da bi bila animacija čim bolj razumljiva in zanimiva za otroke, bomo kot predmet urejanja izbrali Lego kocke in jih urejali po velikosti od najmanjših do največjih.

V zaključnem delu diplomske naloge bomo povzeli bistvo diplomske naloge s poudarkom na lastnem primeru. Poleg tega bomo podali tudi svoje kritično mnenje o nalogi in ideje za izboljšave in nadaljnjo delo.



## 2 Opis računskega razmišljanja

Računsko razmišljanje pri reševanju problemov vključuje tako človeške možgane kot tudi zmogljivosti računalnikov. Sooča se z vprašanji umetne inteligence: »Kaj lahko ljudje storijo bolje kot računalniki? Kaj lahko računalniki naredijo bolje kot ljudje?« Večinoma pa obravnava vprašanje: »Kaj je izračunljivo?« Pri reševanju nekaterih problemov si moramo postaviti tudi vprašanja, kot so: »Kako težko je rešiti problem? Kateri način je najboljši za reševanje tega problema?« Odgovori na ta vprašanja temeljijo na trdnih teoretičnih temeljih [6].

V tem poglavju bomo najprej predstavili začetke uporabe računskega razmišljanja, nato pa si bomo pogledali, kaj vse vključuje računsko razmišljanje. Prav tako bomo našteali nekaj glavnih tehnik in metod, ki se najpogosteje pojavljajo v računskem razmišljanju in za vsako tudi navedli nekaj primerov. Na koncu poglavja bomo navedli nekaj primerov iz vsakdanjega življenja in drugih disciplin.

### 2.1 Začetki

Prvič je izraz računsko razmišljanje uporabil Seymour Papert leta 1996 [4], še pomembnejše mesto pa je dobil zaradi članka Jeannette M. Wing, ki je bil leta 2006 objavljen v ACM Communications [6]. V njem je računsko razmišljanje opisala kot način za »reševanje problemov, načrtovanje sistemov in razumevanje človeškega obnašanja, pri čemer se opira na temeljne koncepte računalništva.« V članku je predlagala, da bi računsko razmišljanje postalo ena od temeljnih spretnosti in znanj vseh, ne samo računalničarjev, podobno kot so branje, pisanje in računanje. Dotaknila se je tudi pomembnosti vključevanja računskih idej v druge discipline. Njeni argumenti so tako navdušili podjetje Microsoft, da je namenilo univerzi Carnegie Mellon subvencijo v vrednosti več milijonov dolarjev za ustanovitev novega centra za preučevanje tega vidika računalništva.



Center računskega razmišljanja je danes lociran na zasebni raziskovalni univerzi Carnegie Mellon. Glavna dejavnost, s katero se ukvarjajo so problemsko usmerjene raziskave (PROBE – angl. Problem oriented Exploration). PROBE razvija in uporablja računalniške koncepte na način, ki nazorno prikazuje pomen računskega razmišljanja, pri tem pa pospešuje osnovne raziskave v računalništvu. Na splošno v PROBE poteka sodelovanje med računalničarji in strokovnjaki iz področja, ki ga je potrebno preučiti. Projekti PROBE običajno potekajo eno leto, v raziskave pa so vključeni tudi sodelavci iz Microsoft Research [8].

## **2.2 Tehnike in metode v računskem razmišljanju**

Računske metode in modeli nas spodbujajo k reševanju problemov in načrtovanju sistemov, ki se jih sami ne bi lotili. Vključuje vrsto spretnosti za reševanje problemov in tehnik, ki jih programerji uporabljajo za pisanje programov, ki so osnova računalniških aplikacij, kot so iskanje, elektronska pošta in zemljevidi. Specifične tehnike računskega razmišljanja zajemajo dekompozicijo problema, razpoznavanje vzorcev, abstrakcijo, oblikovanje algoritmov, analizo podatkov ter vizualizacijo podatkov [12]. V nadaljevanju si bomo poglobljevali nekatere od naštetih tehnik ter navedli nekaj primerov.

### **2.2.1 Dekompozicija**

Dekompozicija je proces, s katerim kompleksen problem ali sistem razčlenimo na dele, ki si jih lažje predstavljamo, jih lažje razumemo, programiramo in vzdržujemo. Dekompozicija problema nas pogosto vodi do prepoznavanja in posploševanja vzorcev in s tem do oblikovanja algoritma [12].

**Primeri:**

- Pri poskušanju neznanе jedi prepoznavamo sestavine v njej, s tem razčlenjujemo jed v posamezne sestavine.
- Ko nekomu opišemo pot do naše hiše, razčlenjujemo pot iz enega konca na drugega.
- V matematiki lahko poljubna števila (npr. 20, 24, 28) razčlenimo na njihove prafaktorje, ki nam pomagajo pri iskanju najmanjšega skupnega večkratnika teh števil:

$$20 = 2 \cdot 2 \cdot 5$$

$$24 = 2 \cdot 2 \cdot 2 \cdot 3$$

$$28 = 2 \cdot 2 \cdot 7$$

najmanjši skupni večkratnik:  $2 \cdot 2 \cdot 2 \cdot 5 \cdot 3 \cdot 7 = 840$

## 2.2.2 Razpoznavanje vzorcev

Razpoznavanje vzorcev je sposobnost prepoznavanja podobnosti ali skupnih razlik, ki nam pomagajo pri napovedovanju rešitev. Prepoznavanje vzorcev je pogosto osnova za reševanje problemov in oblikovanje algoritmov [12].

**Primeri:**

- Otroci prepoznajo vzorce pri svojem obnašanju in reakciji staršev in učiteljev ter tako ugotovijo, kaj je prav in kaj narobe. Njihovo bodoče obnašanje bo temeljilo na teh vzorcih.
- Ljudje iščejo vzorce v tečajih delnic, da se odločijo kdaj jih kupiti in kdaj prodati.
- Ko računamo največjo možno površino pravokotnika z danim obsegom, lahko vidimo vzorce v dolžini, širini ter površini, kot so:
  - ko razlika med dolžino in širino narašča, se površina zmanjšuje,
  - ko se dolžina in širina približujeta v svojih vrednostih, se površina povečuje.

Ti vzorci nas pripeljejo do zaključka, da je pravokotnik z največjo površino kvadrat.

### 2.2.3 Abstrakcija

Proces abstrakcije je najpomembnejši miselni proces računskega razmišljanja. Pri abstrakciji se odločamo, katere podatke bomo izpostavili in katere odmislili. Uporablja se pri iskanju vzorcev, posploševanju primerov in pri parametrizaciji.

Računalništvo uporablja plasti abstrakcije, pri čemer vsaka plast predstavlja drugačen model iste informacije in procesov, vendar pa uporablja množico objektov, ki veljajo le za določen nivo. V računalništvu delamo hkrati z dvema plastema abstrakcije: plast, ki nas zanima, in plast, ki je nad njo ali pa plast, ki nas zanima, in plast pod njo. Dobro opredeljeni vmesniki med plastmi nam omogočajo gradnjo zapletenih sistemov. Vsaka abstraktna višja plast temelji na konkretni nižji plasti. Na primer, abstrakcija na nižji plasti izpostavi podrobnosti strojne opreme računalnika, na katerem teče program, medtem ko se višje plasti ukvarjajo z logiko programa.

#### Primeri:

- Dnevni koledar, ki uporablja abstrakcijo za prikazovanje tedna z vidika dni in ur, nam pomaga pri organizaciji našega časa.
- Zemljevid sveta uporablja abstrakcijo Zemlje z zemljepisno dolžino in širino, kar nam pomaga opisati lokacijo.
- V matematiki pišemo splošne formule s pomočjo spremenljivk namesto števil, tako da jih lahko uporabljamo za reševanje problemov, ki vsebujejo različne vrednosti.

$$(a + b)(a - b) = a^2 - b^2$$

## 2.2.4 Načrtovanje algoritmov

Načrtovanje algoritmov je metoda, pri kateri poskušamo s pomočjo matematičnih postopkov rešiti problem korak za korakom. Načrtovanje algoritmov pogosto temelji na razgradnji problema in iskanju vzorcev, ki nam pomagajo rešiti problem. Eden glavnih ciljev pri načrtovanju algoritmov je, da ustvarimo časovno učinkovit algoritem [12].

### Primeri:

- Ko kuhar napiše recept, s tem ustvari algoritem, ki služi ostalim pri pripravi iste jedi.
- Ko seštevamo in odštevamo ulomke z različnimi imenovalci, sledimo algoritmu:
  - poiščemo najmanjši skupni večkratnik med vsemi imenovalci,
  - pomnožimo števce in imenovalce ulomkov s tisto številko, ki je potrebna, da v imenovalcu dobimo najmanjši skupni večkratnik, ki smo ga določili v prejšnjem koraku,
  - seštejemo ali odštejemo števce, v imenovalcu pa zapišemo najmanjši skupni večkratnik.
- Ko računamo razliko med dvema številoma v odstotkih uporabljamo naslednji algoritem:
  - če je prvotno število večje kot novo, uporabimo:
 
$$\text{odstotno zmanjšanje} = 100 \cdot \frac{(\text{prvotno} - \text{novo})}{\text{prvotno}},$$
  - če je prvotno število manjše kot novo, uporabimo:
 
$$\text{odstotno zvečanje} = 100 \cdot \frac{(\text{novo} - \text{prvotno})}{\text{prvotno}},$$
  - Če nobena izmed zgornjih neenakosti ne velja, sta številki enaki in zato ni spremembe v odstotkih

Te korake lahko s pomočjo programskega jezika zapišemo v algoritem, ki ga bo računalnik izvedel namesto nas. Slika 1 prikazuje program, ki izračuna razliko med dvema številoma v odstotkih, napisan v programskem jeziku Python.

```
prvotna_vrednost = float(input('Vstavi prvotno vrednost: '))
nova_vrednost = float(input('Vstavi novo vrednost: '))
if prvotna_vrednost > nova_vrednost:
    odstotno_zmanjšanje = 100 * (prvotna_vrednost -
                                - nova_vrednost) / prvotna_vrednost
    print 'Odstotno zmanjšanje:', odstotno_zmanjšanje, '%'
elif nova_vrednost > prvotna_vrednost:
    odstotno_zvečanje = 100 * (nova_vrednost -
                                - prvotna_vrednost) / prvotna_vrednost
    print 'Odstotno zvečanje: ', odstotno_zvečanje, '%'
else:
    print 'Ni spremembe v odstotkih.'
```

Slika 1: Program za izračun razlike (v odstotkih) med dvema številoma v programskem jeziku Python.

## 2.3 Računsko razmišljanje v vsakdanjem življenju

Kot smo že omenili računsko razmišljanje vključuje reševanje problemov. Pri tem se izraz problem ne nanaša le na matematične probleme, katerih rešitev je mogoče analizirati s pomočjo dokaza, algoritma ali programa. Opisuje tudi realne probleme, kjer bi rešitve lahko bile v obliki velikih, zapletenih računalniških programov. Marsikdaj so to vsakdanji problemi, katerih rešitev se nam zdi samoumevna, in se pri tem sploh ne zavedamo, da izhajajo iz računalniških konceptov. Poglejmo si nekaj primerov [7]:

- **Načrtovanje:** ko gre otrok zjutraj v šolo, si da v nahrbtnik stvari, ki jih bo ta dan potreboval.
- **Vračanje (ang. backtracking):** ko otrok izgubi rokavice, mu predlagate, da se znova vrne po isti poti nazaj.
- **Cevovod:** dekan je razmišljal, kako bi slovesnost podelitve diplom izpeljal čim hitreje. Z natančno postavitvijo vseh je zasnoval učinkovit cevovod. Prodekan je prebral ime diplomanta in njegove zasluge, diplomant je prejel svojo diplomu, stisnil roko in se slikal. Ta cevovod je omogočil neprekinjeno prihajanje diplomantov na oder.

- **Razpršitev:** Profesor je razlagal o razprševalni funkciji, ki jo njegovi otroci uporabljajo doma pri shranjevanju Lego kock. Shranjujejo jih v več različnih kategorijah: pravokotne debele kocke, nepravokotne debele kocke, tanke (vseh oblik), osi, kovice in distančniki, tisto kar paše na osi, krogle in priključki ter raznovrstne. Imajo tudi pravila za razvrstitev delčkov, ki spadajo v več kategorij hkrati. Čeprav je način precej preprost, prihrani veliko časa, pri iskanju posameznega elementa.
- **Urejanje:** Profesor računalništva in profesionalni trobentač je nekoč povedal tole zgodbo: »Prišel sem na nastop z orkestrom in vodja ansambla nam je dal note za 200 pesmi in določil seznam 40 naslovov, ki smo si jih morali pripraviti. Vsi so začeli iskati pesmi in jih zlagati na kup eno po eno. Odločil sem se, da bom skladbe najprej uredil po abecednem vrstnem redu in nato poiskal pesmi, ki jih potrebujem. Sam sem še vedno urejal pesmi, ko so bili nekateri že na polovici in s tem sem si zaslužil nekaj začudenih pogledov. Vendar sem na koncu končal pred vsemi ostalimi. Temu se reče računsko razmišljanje«.

## 2.4 Računsko razmišljanje v drugih disciplinah

Računsko razmišljanje vpliva na raziskovanje v skoraj vseh disciplinah, tako v naravoslovnih kot tudi humanističnih vedah. Začelo se je z uporabo računskega modeliranja in simuliranja, danes pa uporabljamo podatkovno rudarjenje in strojno učenje za analizo ogromnih količin podatkov. Zato je računanje priznано kot tretji steber znanosti, poleg teorije in eksperimentiranja. V nadaljevanju bomo predstavili vplive računskega razmišljanja na posameznih področjih.

### 2.4.1 Kemija

Zahvaljujoč napredkom v metodologiji in računalniški zmogljivosti je računanje pri raziskovanju in ponazarjanju kemijskih pojavov ter zagotavljanje podatkov, ki se jih ne da pridobiti iz poskusov, postalo že rutina. Danes večina kemijskih publikacij vključuje izračune, ki so bili nekaj let nazaj obravnavani kot področje teoretikov in

visokozmogljivih računalnikov. Kljub temu to ne pomeni, da so vsi problemi rešljivi. Optimizacija in preiskovalni algoritmi prinašajo eksperimentalni kemiji najboljše kemikalije za posamezne namene ali za izboljšanje reakcij. Metode iz informatike, ki temeljijo na velikih bazah poskusnih in drugih podatkov, se uporabljajo za raziskave zdravil [17].

Pomembna veja v kemiji je računska kemija, ki uporablja načela računalništva za pomoč pri reševanju kemijskih problemov. Rezultati teoretične kemije se uporabljajo v zmogljivih računalniških programih za izračun strukture in lastnosti molekul in snovi. Rezultati izračunov običajno dopolnjujejo eksperimentalno dobljene podatke, v določenih primerih pa lahko napovejo tudi prezrte kemijske pojave. Računska kemija se pogosto uporablja pri snovanju novih zdravil in materialov. Računske metode se uporabljajo za izračun tako statičnih (struktura sistema v osnovnem stanju), kot tudi dinamičnih kemijskih problemov (struktura sistema med kemijsko reakcijo). V vseh primerih procesorski čas in ostali viri (pomnilnik, prostor na trdem disku) narašča zelo hitro z velikostjo obravnavanega sistema. Sistem je lahko ena sama molekula, skupek molekul ali celo trdnina. Računske metode v kemiji so glede natančnosti zelo različne: od zelo natančnih do precej približnih, pri čemer se za izbiro metode običajno odločimo na podlagi velikosti problema [9].

## 2.4.2 Biologija

Računalništvo in biologija se prepletata že desetletja. Biologi se pri analizi velikih količin podatkov zanašajo na računalniške metode, hkrati pa veliko računskih metod temelji na načelih bioloških sistemov. V zadnjih letih sta se ti dve smeri zelo zblížali. Sekvenciranje genoma je zelo odvisno od algoritmov za pospeševanje analize podatkov. Eden takih algoritmov je tudi »shotgun« algoritem, ki je pripomogel k razvozlanju človeškega genoma. Abstrakcije nam pomagajo pri predstavitvi dinamičnih procesov, ki jih najdemo v naravi, kot so celični cikel ali zlaganje proteinov. Obseg in stopnja, na kateri znanstveniki in inženirji zdaj zbirajo in pripravljajo podatke – s pomočjo instrumentov, eksperimentov in simulacij – so zahtevali napredek v podatkovni analizi, shranjevanju in iskanju, pa tudi v vizualizaciji podatkov [7].

### 2.4.3 Novinarstvo

O tem, da mora biti novinar digitalno pismen, ni nobenega dvoma. Pri zbiranju in objavljanju novic sta potrebni sposobnost pisanja programov, ki brskajo po javnih evidencah in oblikovanja vmesnikov, ki naredijo informacije zanimive, ustrezne in dostopne. To zahteva sposobnosti in znanje s področja programiranja in oblikovanja, saj je potrebno ustvariti interaktivne predstavitve. Prav tako je potrebno obvladati socialne medijske tehnologije, ki se uporabljajo za organiziranje spletnih skupnosti. Vendar pa to ne pomeni, da morajo novinarji postati programerji. Morajo biti le sposobni abstraktno predstaviti delo, ki ga opravljajo, razumeti široko paleto računalniških orodij, ki jih imajo na voljo, in sodelovati pri uporabi teh orodij. To pomeni, da morajo razumeti temeljne strukture in procese medijskega ustvarjanja [5].

### 2.4.4 Igre

Raziskave so pokazale, da igre pomagajo pri razvoju sposobnosti računskega razmišljanja otrok, pri tem pa je logično razmišljanje za njih izziv in zabava. Mnogi časopisi imajo logične uganke, kot so sudoku in kakuro. Pri teh ugankah moramo s pomočjo sklepanja iz nekaj podanih vrednosti, s številkami zapolniti vsa polja. Na spletu je tudi veliko iger, s katerimi se otroci lahko naučijo osnov programiranja [11]. Primer take igre je RoboZZle, ki jo bomo predstavili v nadaljevanju.

Prav tako obstajajo številni prosto dostopni programi za enostavno kreiranje iger. Eden izmed takih programov je Scratch, ki je bil poimenovan kot »YouTube interaktivnega medija«. Vsak dan ljudje iz celega sveta na strani objavijo več kot 1500 novih projektov, z izvorno kodo, ki je prosto dostopna. Zbirka projektov na strani je zelo raznolika. Najdemo razne video igre, interaktivna glasila, znanstvene simulacije, virtualne ogleda, rojstnodnevne čestitke, animirana plesna tekmovanja in interaktivne vadnice; vse je napisano v Scratch okolju. Večina ljudi na spletni strani je stara med 8 in 16 let, čeprav sodeluje tudi precej odraslih. S tem, ko otroci programirajo in si delijo Scratch projekte, se naučijo osnov računalniških in matematičnih konceptov, hkrati pa se naučijo pomembnih strategij za projektiranje, reševanje problemov in sodelovanje. Cilj izumiteljev Scratcha ni bil pripraviti ljudi na kariero poklicnih programerjev, ampak ustvariti novo generacijo kreativnih, sistematičnih mislecev, ki z lahkoto uporabljajo programiranje za izražanje svojih idej. Scratch podpira učenje računskega



razmišljanja, sklop konceptov, praks in vidikov, ki črpajo ideje iz računalniškega sveta [20].

### 2.4.5 Univerzitetno izobraževanje

Računsko razmišljanje ima svoj pomen tudi v izobraževanju. Univerzitetna središča po svetu so ponovno preučila učne načrte v dodiplomskem izobraževanju na področju računalništva. Na univerzi Carnegie Mellon so uvedli predmete za spodbujanje računskega razmišljanja. Računsko razmišljanje se pojavlja v matematiki, biologiji, kemiji, oblikovanju, ekonomiji, financah, jezikoslovju, mehaniki, fiziki in statistiki. Imajo celo tečaj računske fotografije, programe za računalniško glasbo, organizacije in družbo [7]. V tabeli 1 je prikazano, katere discipline so združene v inštitutih na univerzi Carnegie Mellon.

<b>INŠTITUT</b>	<b>DISCIPLINE KI JIH ZDRUŽUJE</b>
<b>The Robotics Institute</b>	računalništvo, elektrotehnika, strojništvo
<b>The Language Technologies Institute</b>	računalništvo, jezikoslovje
<b>The Human – Computer Interacion Institute</b>	računalništvo, oblikovanje, psihologija
<b>The Machine Learning Department</b>	računalništvo, statistika
<b>The Institute for Software Research</b>	računalništvo, javni red, družbene vede
<b>The Lane Center for Computational Biology</b>	računalništvo, biologija
<b>The Entertainment Technology Center</b>	računalništvo, igrilstvo

Tabela 1: Discipline inštitutov, na področju računalništva na univerzi Carnegie Mellon.

### 2.4.6 Izobraževanje v K - 12

Prepozno je, da bi se z osnovnimi koncepti računalništva seznanili šele na fakultetah. Mnogi bomo delali na področjih, ki vključujejo ali pa so pod vplivom računalništva. Zato moramo začeti z algoritmičnim reševanjem problemov in računalniškimi metodami že na začetku šolanja. Računalniško izobraževanje lahko igra veliko vlogo pri algoritmičnem reševanju praktičnih problemov in pri vključevanju uporabe računalniških metod in orodij na različnih področjih. Otroci se že na nižji stopnji učijo,

kako razmišljati in reševati probleme, vendar pa lahko računalničarji pomagajo učiteljem pri razumevanju procesov in algoritmov ter poskušajo ugotoviti, kje lahko računalniki koristijo pri računanju in manipulaciji s podatki [14].

Ko je Jeanette Wing napisala, da »računsko razmišljanje predstavlja splošno uporaben pristop in znanje za vsakogar« je spodbudila inovacije, ki bi lahko služile za doseg tega cilja. Za sprejetje ideje računskega razmišljanja na ravni ZDA je bil potreben bolj sistemski pristop. Zato je Nacionalna ustanova za znanost (NSF - angl. The National Science Foundation) pozvala k sodelovanju Mednarodno združenje za tehnologijo v izobraževanju (ISTE - angl. The International Society for Technology in Education) in Društvo učiteljev računalništva (CSTA - angl. The Computer Science Teachers Association). Poleti leta 2009 sta CSTA in ISTE začela večfazni projekt namenjen razvoju učnega programa računskega razmišljanja za K-12. Ti dve organizaciji sta še posebej primerni za to, saj zelo dobro poznata K-12, poleg tega pa imata veliko strokovnega znanja s področja razvoja izobraževanih standardov, učnih materialov in poklicnega razvoja učiteljev. Začeli so z načelom, da bi vsak udeleženec izobraževanja do konca srednje šole moral poznati osnove računskega razmišljanja. To bi prineslo računsko razmišljanje v uradno izobraževanje in učitelji na vseh stopnjah in na vseh vsebinskih področjih bi prispevali k spretnostim s področja računskega razmišljanja, ki jih pridobi učenec. Lotili so se projekta z naslovom Leveraging Thought Leadership for Computational Thinking in K-12 Curriculum.

Izobraževanje v K-12 je zelo kompleksno politično okolje, kjer se mnoge konkurenčne prioritete, ideologije in pedagoške strategije borijo za prevlado. Vsako prizadevanje za doseg sprememb v tem okolju zahteva podrobno razumevanje dejanskega stanja sistema. Za vključevanje računskega razmišljanja v K-12 je zato potreben praktičen pristop, ki je utemeljen z operativnimi definicijami. Osredotočili so se na vrsto vprašanj, ki so povezana z izvajanjem v K-12:

- Kako bi bilo računsko razmišljanje videti v razredu?
- Katera znanje bi študentje s tem dokazali?
- Kaj bi potrebovali učitelji, da bi računsko razmišljanje lahko praktično izvedli?
- Kaj od tega, kar učitelji delajo že zdaj, se da spremeniti in razširiti?

Aprila 2010 so se zbrali predstavniki različnih šol, da bi skupaj zapisali operativno definicijo računskega razmišljanja za izobraževanje K-12. Definicija, ki so jo potrdili s pomočjo raziskav učiteljev, računalničarjev, vzgojiteljev in raziskovalcev se glasi:

Računsko razmišljanje je proces reševanja problemov, ki vključuje naslednje lastnosti:

- formuliranje problemov na način, ki nam omogoča, da pri reševanju uporabljamo računalnik in ostala orodja,
- logična organizacija in analiza podatkov,
- prikazovanje podatkov s pomočjo abstrakcij, kot so modeli in simulacije,
- avtomatizacija rešitev skozi algoritmično razmišljanje,
- posploševanje in prenos procesa reševanja problemov na najrazličnejše probleme.

### 2.4.7 Ostala področja

Računsko razmišljanje vpliva tudi na discipline in poklice izven znanosti in inženirstva, npr. algoritmično medicino, računsko arheologijo, računsko ekonomijo, računske finance, računanje in novinarstvo, računsko pravo, računske družbene vede in digitalne humanistične vede. Podatkovna analitika se uporablja v usposabljanju vojaških novincev, pri odkrivanju nezaželene elektronske pošte in goljufijah s kreditnimi karticami. Mnoge znanosti in inženirske discipline uporabljajo ogromne računalniške simulacije matematičnih modelov ali fizikalnih procesov. Letalska in vesoljska industrija se zanašata na simulacijo celotnih letal ali vesoljskih misij. Digitalne knjižnice, zbirke in predmeti s področja humanistike in umetnosti z metodami, kot je podatkovno rudarjenje, dobivajo možnost za odkrivanje novih trendov, vzorcev in povezav v razumevanju človeštva. Računsko razmišljanje je preoblikovalo statistiko, kjer lahko s pomočjo strojnega učenja, avtomatizacije Bayesovih metod in uporabe grafičnih verjetnostnih modelov, prepoznavamo vzorce in poiščemo nepravilnosti v obsežnih podatkovnih bazah, v astronomskih kartah, magnetni resonanci in napravah, ki skenirajo kreditne kartice. Statistični oddelki danes zaposlujejo tudi računalničarje, računalniške fakultete pa odpirajo nove oddelke za statistiko [7].

Številne discipline učijo reševanja problemov, ter spodbujajo k logičnemu in algoritmičnemu razmišljanju. Računalničarji lahko pomagajo pri razumevanju tega, kako računalniške koncepte uporabiti pri problemih na drugih področjih, kot je prikazano v tabeli 2.

Koncepti računskega razmišljanja	Računsko razmišljanje	Matematika	Naravoslovje	Družbene vede	Jezikoslovje
<b>Zbiranje podatkov</b>	Najti vir podatkov za problemsko območje	Najti vir podatkov za problemsko območje, (npr. metanje kovancev ali kock)	Zbiranje podatkov iz poskusov	Preučevanje statističnih ali populacijskih podatkov	Jezikovna analiza stavkov
<b>Analiza podatkov</b>	Pisanje programov za osnovne statistične izračune	Štetje pojavitev grba ali cifre pri kovancih, metov kock in analiza rezultatov	Analiziranje podatkov iz poskusov	Prepoznavanje trendov iz statističnih podatkov	Prepoznavanje vzorcev za različne tipe stavkov
<b>Predstavitev podatkov</b>	Uporaba podatkovnih struktur kot so tabela, sklad, vrsta, graf, razpršena tabela	Uporaba diagramov za predstavitev podatkov; uporaba množic za obvladovanje podatkov	Povzemanje podatkov iz poskusov	Povzemanje in predstavljanje trendov	Predstavitev vzorcev iz različnih tipov stavkov
<b>Problem razgradnje</b>	Definiranje metod in funkcij	Uporabljanje vrstnega reda operacij v izrazih	Izdelava klasifikacij ali taksonomij		Pisanje orisa
<b>Algoritmi in procedure</b>	Preučevanje klasičnih algoritmov; izvajanje algoritma	Deljenje, množenje, seštevanje in odštevanje	Testiranje		Pisanje navodil

Koncepti računskega razmišljanja	Računsko razmišljanje	Matematika	Naravoslovje	Družbene vede	Jezikoslovje
<b>Abstrakcija</b>	Uporaba postopkov, ki zaobjamejo množico pogostih ukazov za izvajanje funkcij; uporaba pogojnih skokov, zank, rekurzij...	Uporaba spremenljivk v algebri; prepoznavanje bistvenih dejstev problema; primerjava funkcij iz algebre s funkcijami iz programiranja	Gradnja vzorca na podlagi fizikalnih dejstev	Povzemanje dejstev; sklepanje iz dejstev	Uporaba komparacije in metafore; pisanje zgodb z odstavki

Tabela 2: Uporaba računalniških konceptov na področju računskega razmišljanja, matematike, naravoslovja, družbenih ved in jezikoslovja [1].

## 3 Konkretni primeri iz posameznih področij

Predstavili bomo nekaj primerov, pri katerih lahko najdemo vpliv tehnik in metod, ki jih vključuje računsko razmišljanje. S področja robotike bomo pogledali robotsko kirurgijo. Pomemben algoritem, ki je imel velik vpliv pri projektu človeškega genoma, je algoritem »shotgun«. Predstavili bomo tudi enega izmed projektov PROBE na univerzi Carnegie Mellon in sicer izmenjavo ledvic. Kot smo že omenili, imajo igre velik vpliv pri miselnem razvoju otrok, zato bomo predstavili igro RoboZZle, s pomočjo katere se otroci naučijo osnov programiranja. Kot zadnjo si bomo pogledali eno izmed lekcij pri poučevanju računalništva v K – 12, ki otrokom pomaga spoznati algoritem in jih uči algoritmičnega razmišljanja.

### 3.1 Robotska kirurgija

Robotska kirurgija, računalniško podprta kirurgija in robotsko podprta kirurgija so danes povsem običajni pojmi pri kirurških posegih. Robotsko podprta operacija je bila razvita za minimalno invazivno (laparoskopsko) kirurgijo in za povečanje zmogljivosti kirurgov, ki opravljajo odprte operacije.

V primeru robotsko podprte minimalne invazivne kirurgije, kirurg namesto neposredne uporabe instrumentov, uporablja enega od dveh načinov za obvladovanje instrumentov [18]:

- *neposredno upravljanje robotskih rok*: kirurg s pomočjo robotskih rok opravlja običajne kirurške gibe.
- *z računalniškim nadzorom*: kirurg uporablja računalnik za krmiljenje robotskih rok in stikal. Ena od prednosti uporabe tega načina je, da se lahko kirurg nahaja kjerkoli na svetu, kar omogoča oddaljene operacije.

V primeru večje odprte operacije avtonomni instrumenti nadomeščajo običajne instrumente iz jekla, za določene dejavnosti (kot je širjenje reber) z bolj gladkimi in

kontroliranimi gibi, kot bi jih lahko opravljala človeška roka. Glavni cilj teh instrumentov je zmanjšanje ali odprava poškodb tkiva, ki se običajno pojavijo pri odprtih operacijah, pri tem pa usposabljanje kirurga za tovrstno delo ne traja več kot nekaj minut. Ta pristop je najpogostejši pri operacijah srca in prsnega koša. Robotska kirurgija uporablja Da Vinci robotski sistem v kirurgiji. Sistem sestavljajo tri komponente [15]:

- kirurška konzola,
- robot s štirimi rokami, ki ga upravlja kirurg (en roka je za nadzor kamere, preostale tri pa za upravljanje s instrumenti),
- sistem za prenos tridimenzionalne slike.

Robot zazna gibe rok kirurga in jih v elektronski obliki prevaja v pomanjšane mikro premike, s katerimi upravlja z drobnimi instrumenti. Prav tako zazna in izloči vse tresljaje, ki jih povzročijo roke kirurga, tako da se robotsko ne podvajajo. Kamera, ki se uporablja v sistemu, zagotavlja povečan tridimenzionalen posnetek. Da Vinci sistem je ameriški vladni urad za zdravila in prehrano (FDA - angl. Food and Drug Administration) odobril za različne kirurške posege, npr. za operacijo raka prostate, histerektomije in mitralna popravila, in se uporablja v več kot 800 bolnišnicah v Ameriki in Evropi.

#### **PREDNOSTI ROBOTSKE OPERACIJE:**

- natančnost,
- manj brazgotin,
- manjša izguba krvi,
- manj bolečin,
- hitrejše okrevanje,
- artikulacija preko običajne manipulacije,
- tridimenzionalna povečava,
- krajša hospitalizacija,
- manjša izguba krvi s transfuzijami,
- manjša uporaba protibolečinskih tablet.

**SLABOSTI ROBOTSKE OPERACIJE:**

- visoki stroški,
- potrebno dodatno usposabljanje kirurgov (med usposabljanjem lahko operacija traja dvakrat dlje kot običajna operacija, kar vodi do problemov s prostori in daljšo anestezijo).

## 3.2 Shotgun algoritem

Eden od mejnikov razvoja na področju biologije je bil gotovo projekt človeškega genoma, ki je bil največji in najdražji projekt v zgodovini biologije. Projekt se je začel leta 1990 in naj bi trajal 15 let, vendar je bil zaključen v manj kot 13 letih. V projektu je sodelovalo več sto znanstvenikov, ki so delali v 20 centrih v Nemčiji, na Kitajskem, v ZDA, Franciji, Veliki Britaniji ter na Japonskem. Namen projekta je bil določitev zaporedja nukleotidnih baz (adenin, gvanin, citozin, timin) v genskem zapisu. Rezultati so omogočili vpogled v katalog vseh genov in njihovih proteinov, ki sodelujejo pri delovanju človeških celic. Razvozlanje genoma omogoča revolucijo v medicini, še zlasti pri ugotavljanju genetsko pogojenih bolezni in njihovem zdravljenju še pred rojstvom, pa tudi pri razumevanju bolezni.

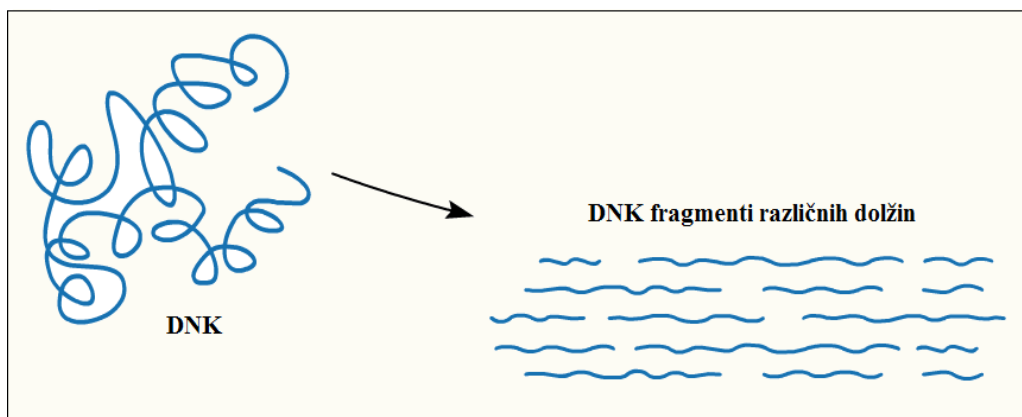
Človeški genom je celotna dedna zasnova posameznika, ki vsebuje vse podatke, ki so potrebni za sintezo beljakovin v človeškem organizmu. Zajema 23 parov kromosomov ter mitohondrijsko DNK. Človeški genom je sestavljen iz okoli 3,4 milijarde baznih parov nukleotidov in nosi zapis za okoli 20.000 - 25.000 genov. Genom vsakega posameznika, razen enojajčnih dvojčkov, je edinstven, zato je potrebna ugotovitev zaporedij različnih variacij posameznih genov. Določanje tega zaporedja je bilo končano leta 2003. Eden od predhodnikov tehnologije, s katero so razvozlati človeški genom, je bil »shotgun« sekvenciranje.

Ta metoda vključuje razbitje ciljnega genoma na majhne, prekrivajoče se fragmente, kloniranje in sekvenciranje vseh fragmentov in uporabo računalniških analiz pri iskanju prekrivajočih se ali sosednjih sekvenc v vsakem fragmentu. Potek »shotgun« metode bomo povzeli v nekaj korakih [13]:

1. Izolacija DNK iz ciljnega organizma. DNK se nato naključno razreže na različno dolge fragmente, kar prikazuje slika 2. To je potrebno storiti zato, ker ni

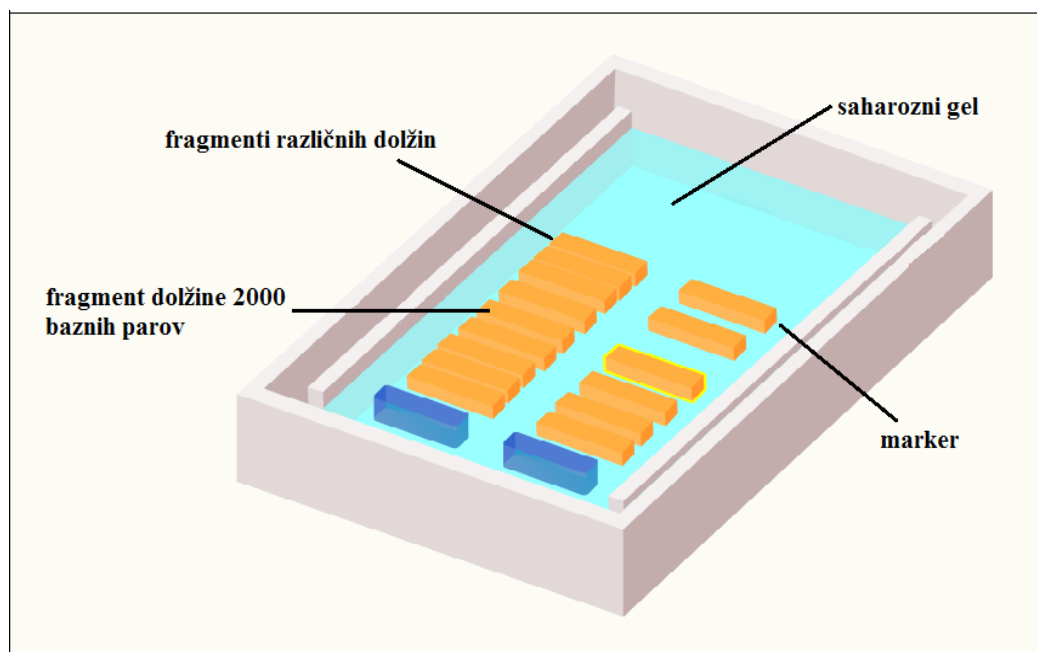


mogoče sekvencirati večmilijonskih verig DNK. Povprečna in zaželena dolžina DNK fragmenta je 2000 baznih parov.



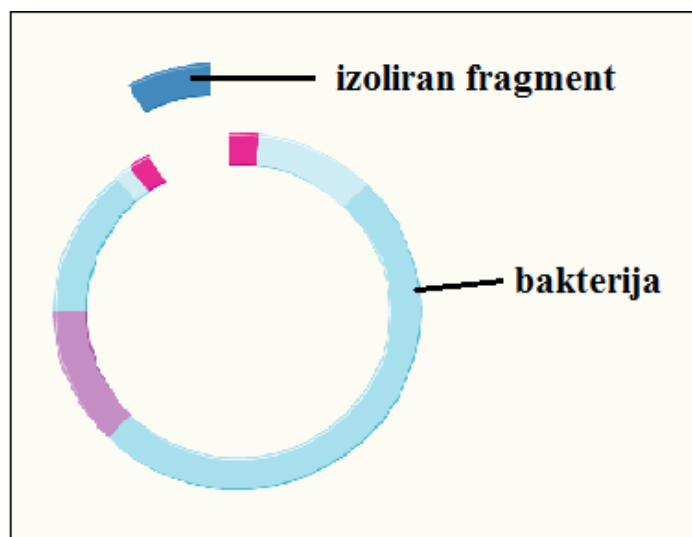
Slika 2: Razrez DNK na različno dolge fragmente [13].

2. Razrezana DNK se naloži na saharozni gel. S pomočjo markerja, ki je dolg 2000 baznih parov, se poišče tisti fragment, ki je dolg toliko kot marker. Najden fragment se nato izolira, saj je potreben za nadaljnje delo. Ta korak prikazuje slika 3.



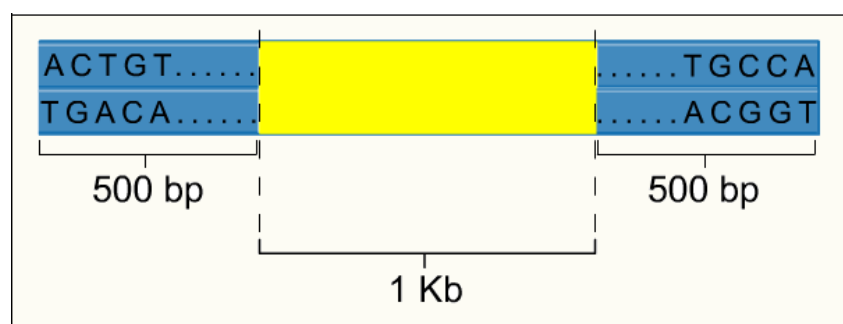
Slika 3: Iskanje fragmenta dolžine 2000 baznih parov [13].

- Izolirani fragment se nato klonira s pomočjo bakterij. To se naredi tako, da se bakterija s pomočjo encima razpre na točno določenem mestu, izoliran fragment pa se pripne na bakterijo. Postopek je prikazan na sliki 4.



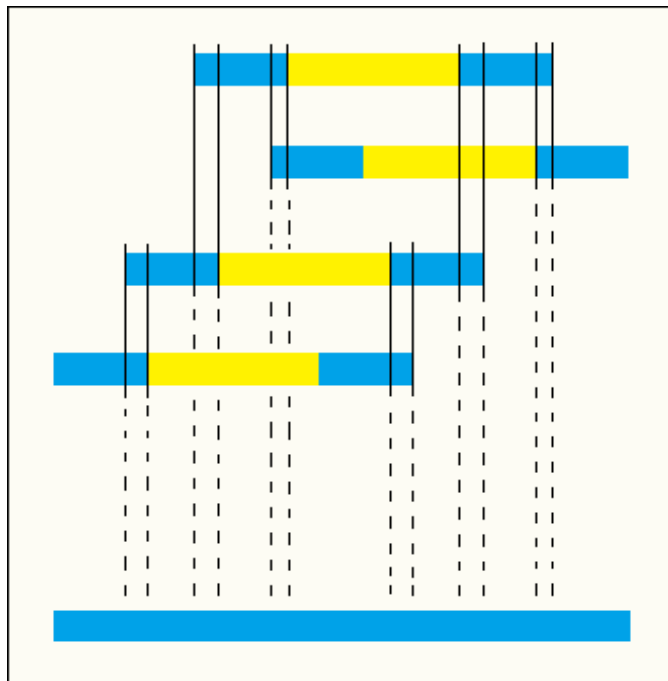
Slika 4: Kloniranje fragmenta s pomočjo bakterije [13].

- S pomočjo Sangerjeve metode za sekveniciranje DNK, se na vsakem koncu kloniranih fragmentov določi 500 baznih parov. To prikazuje slika 5.



Slika 5: Fragment, kjer je na vsakem koncu določenih 500 baznih parov nukleotidov [13].

5. Na koncu se z uporabo računalniških algoritmov poiščejo prekrivajoči se fragmenti in se združijo v eno dolgo verigo. Združevanje fragmentov v verigo prikazuje slika 6.



Slika 6: Iskanje prekrivajočih se koncev fragmentov in združitvev v eno verigo.

V tem primeru lahko zasledimo več vidikov računskega razmišljanja:

- posebna pojem natančno oblikovanega in nedvoumnega postopka, ki se uporablja vedno znova in znova,
- iskanje,
- ujemanje vzorcev,
- iterativno izpopolnjevanje,
- naključnost, kot sredstvo pri ponavljajočih delitvah.

### 3.3 Optimalna izmenjava ledvic

Kaj bi se zgodilo, če bi kirurgi in zdravniki uporabljali računsko razmišljanje za sprejemanje odločitev pri presejevanju organov? Ali je mogoče optimizirati dodeljevanje organov tako, da bi bilo rešenih več ljudi?

Samo v ZDA zaradi smrtonosne odpovedi ledvic vsako leto zboli preko 30.000 ljudi. Potrebe daleč presegajo število ledvic, ki so primerne za darovanje. Za presaditev pridobimo ledvice večinoma od umrlih darovalcev. Ker je ledvica parni organ, lahko eno od obeh ledvic daruje tudi živi darovalec, ki je z bolnikom genetsko, družinsko ali emocionalno povezan. Na žalost pa je verjetnost, da se tipa krvi in tkiva pacienta in donatorja ujemata, zelo majhna. To odpira vrata izmenjavi ledvic.

Predstavljajmo si dva para donator – pacient, ki ju bomo zaradi lažjega razumevanje označili kot par A in par B. Pacient A, ki je na čakalni listi za ledvico, najde prijatelja ali sorodnika, donatorja A, katerega ledvica zanj sicer ni primerna, vendar bi donator A lahko daroval organ pacientu B, donator B pa bi lahko daroval pacientu A. S tem se ustvari cikel dolžine dve. Tudi nekoliko daljši cikli so še vedno praktični, vendar pa so raziskave pokazale, da je najdaljši še optimalni cikel, cikel dolžine 4. Treba se je namreč zavedati, da je vse operacije potrebno izvesti hkrati, zato bi pri večjih ciklih lahko nastala stiska s prostorom in osebjem, ki sodeluje pri operacijah.

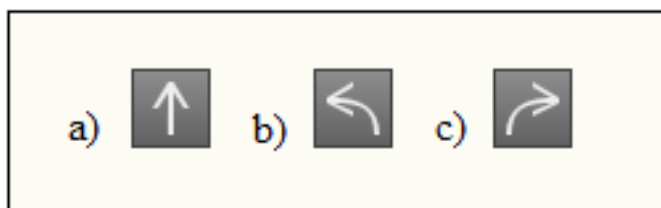
Članek »Kliring algoritem za trge z blagovno menjavo: omogočanje mednarodne izmenjave ledvic« [2] opisuje optimalni algoritem, ki se uporablja pri problemu izmenjave. V članku so poskušali ugotoviti, koliko je še optimalna dolžina cikla, saj ta ne sme biti predolg, ker morajo biti vse operacije izvedene hkrati. Dokazali so, da je kliring problem z omejitvijo dolžine cikla NP - poln. To je eden glavnih izzivov pri vzpostavitvi nacionalne izmenjave ledvic. V članku je predstavljen prvi algoritem, ki lahko deluje na državni ravni. Opisani algoritem optimalno reši problem izmenjave ledvic z 10.000 pari. Predhodni pristopi niso delovali na primerih, ki so vsebovali več kot 900 parov, saj je zmanjkalo pomnilnika. Nadaljnje raziskave na tem področju še potekajo. Osredotočajo se na iskanje hitrejšega algoritma [8].

## 3.4 Igra RoboZZle

Miselne igre lahko spodbujajo logično in računsko razmišljanje, ki koristi tudi pri reševanju problemov v vsakdanjem življenju. RoboZZle ja igra, s katero se lahko naučimo programiranja. S pomočjo treh ukazov in uporabo programskih konceptov, kot so funkcije, pogoji in rekurzija lahko robot reši problem. Pri tej igri robota predstavlja raketa, ki se glede na ukaze, ki smo jih zapisali, premika po poljih in pobira zvezdice. Igra je končana, ko na igralni ploščadi ni nobene zvezdice več, pri tem pa raketa ne sme zaviti iz označenih polj.

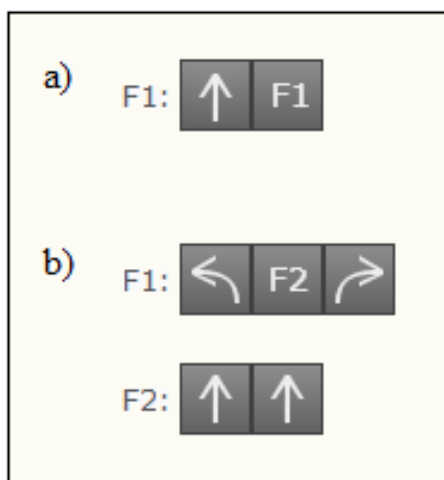
### 3.4.1 RoboZZle ukazi

- a) **UKAZI ZA PREMIKANJE:** ukazi, prikazani na sliki 7 omogočajo, da se robot premika naprej, se obrne za 90 stopinj v levo ali desno.



Slika 7: Ukazi za premikanje robota po igralni plošči: a) premik za eno polje naprej, b) obrat za 90 stopinj v levo, c) obrat za 90 stopinj v desno [19].

- b) **UKAZI ZA FUNKCIJE:** niz ukazov lahko uporabimo večkrat, tako da jih opredelimo kot funkcijo. Pri tem nam pomagajo ukazi, ki so prikazani na sliki 8. Funkcija se izvaja takole: ko pokličemo funkcijo, se izvajanje na trenutni točki shrani na sklad in ko se poklicana funkcija konča, se izvajanje nadaljuje od shranjene točke naprej. Shranjevanje poteka na skladu, saj funkcija lahko kliče drugo, ta naslednjo in tako dalje.



Slika 8: Ukazi za klic funkcije: a) funkcija pokliče samo sebe, b) klic druge funkcije [19].

- c) **POGOJNI UKAZI:** ukazi, ki veljajo samo pod določenim pogojem. Če pogoj ni izpolnjen, se ukaz preskoči. Primer pogojnega ukaza je prikazan na sliki 9.

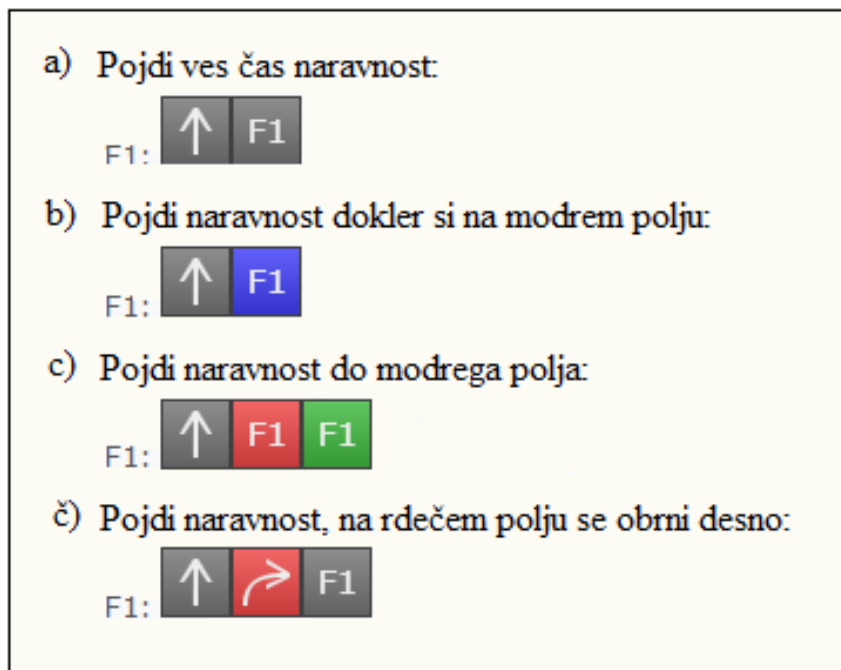


Slika 9: Pogojni ukaz, ki se izvrši, če se nahajamo na modrem polju.

### 3.4.2 Rekurzivne funkcije

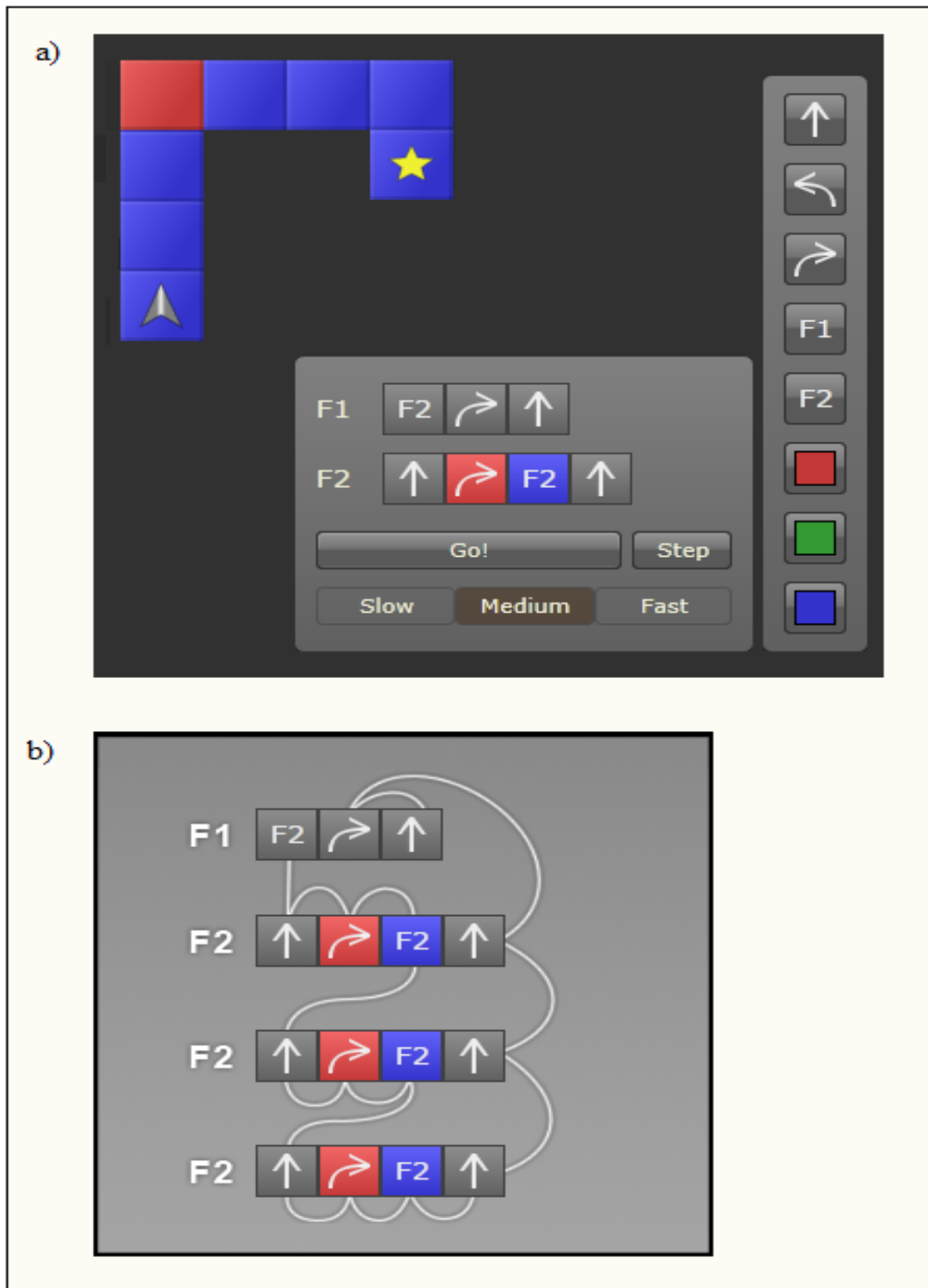
Če funkcija kliče sama sebe, pravimo, da je funkcija rekurzivna. Pri igri RoboZZle poznamo dva osnovna načina uporabe rekurzije. Pri prvem načinu nastanejo zanke, pri drugem, bolj kompleksnem, pa se stvari shranjujejo na sklad.

- a) **UPORABA ZANKE:** Če želimo dobiti zanko, moramo kodo, ki jo želimo ponavljati, postaviti na začetek funkcije, ki sledi rekurzivnemu klicu. Kadar ima rekurzivni klic pogoj, mu pravimo ustavitveni pogoj. Ko ustavitveni pogoj ni izpolnjen, se klic vrne in se izvede koda za rekurzivnim klicem. Uporaba rekurzivne funkcije in nastanek zanke je prikazana na sliki 10.



Slika 10: Prikaz rekurzivnih funkcij: a) robot se premika ves čas za eno mesto naprej; b) če je robot na modrem polju, se premakne za eno mesto naprej; modro polje predstavlja ustavitveni pogoj; c) robot se premika naprej, dokler ne pride do modrega polja; č) robot se premika naprej in se na rdečem polju obrne za 90 stopinj v desno [19].

- b) UPORABA SKLADA:** med izvajanjem programa se podatki shranjujejo na sklad. Sklad je podatkovna struktura, ki deluje po načelu zadnji noter, prvi ven (LIFO – angl. Last In, First Out). To pomeni, da podatek, ki ga nazadnje shranimo na sklad, najprej preberemo z njega. V našem primeru se na sklad shranjujejo poklicane funkcije. Vsaka funkcija se enkrat konča, ker smo pobrali že vse zvezdice ali pa ker se je zgodila neveljavna operacija. Če naredimo kup klicev, preden se je neka funkcija zaključila, se bodo najprej izvršile poklicane. Primer uporabe sklada in njegovo izvajanje je prikazano na sliki 11.



Slika 11: Izvajanje rekurzivne funkcije z uporabo sklada: a) problem, ki ga moram rešiti in končna rešitev; b) potek izvajanja [19].



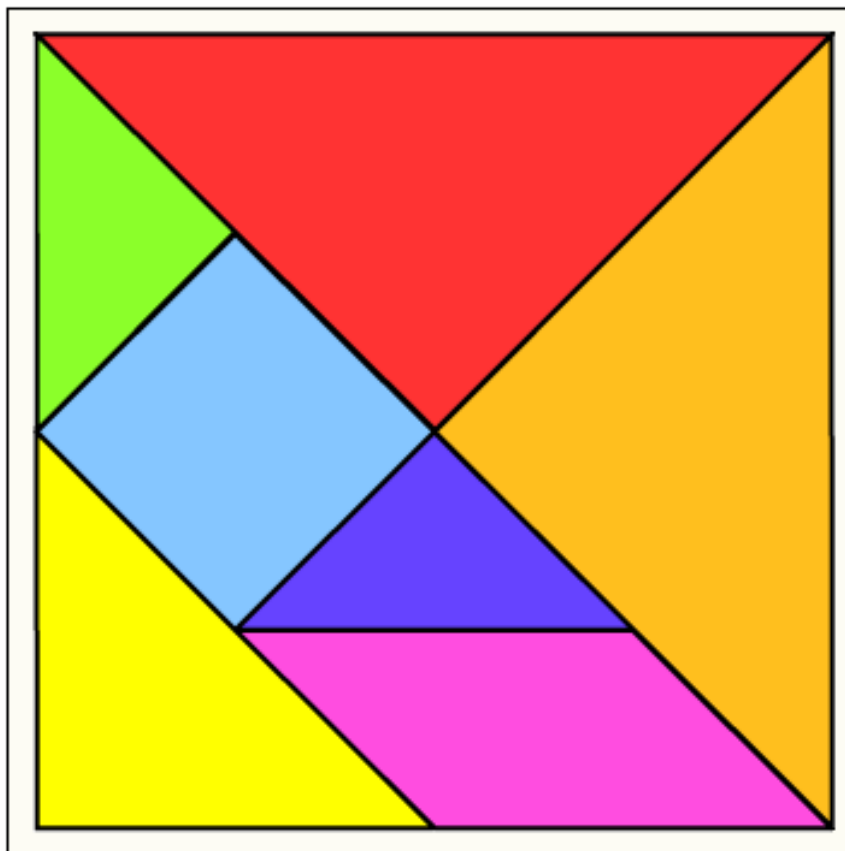
## 3.5 Kaj je algoritem in kaj algoritmično razmišljanje za otroke

Ker ima računsko razmišljanje velik vpliv v izobraževanju, si bomo pogledali primer lekcije, ki jo lahko najdemo na strani Exploring Computational Thinking [12]. Namen lekcije je otroke seznaniti s pojmom algoritma in algoritmično mišljenje in jim to predstaviti na konkretnem primeru.

Za začetek bomo le teoretično predstavili, kaj je algoritem in kaj algoritmično razmišljanje. Algoritem je metoda za reševanje problema, ki je sestavljena iz točno določenih navodil. Algoritmično razmišljanje je razmišljanje o tem, kako doseči cilj. Je tudi natančno in usmerjeno razmišljanje o metodah, nabor med seboj povezanih sposobnosti, s katerimi gradimo in razumemo algoritme:

- sposobnost analiziranja danega problema,
- sposobnost natančnega določanja problema,
- sposobnost iskanja osnovnih akcij, ki so primerne za reševanje problema,
- sposobnost gradnje pravilnega algoritma za dani problem s pomočjo osnovnih akcij,
- sposobnost razmišljanja o vseh možnih posebnih in običajnih primerih, ki se lahko pojavijo pri problemu,
- sposobnost izboljšanja učinkovitosti algoritma.

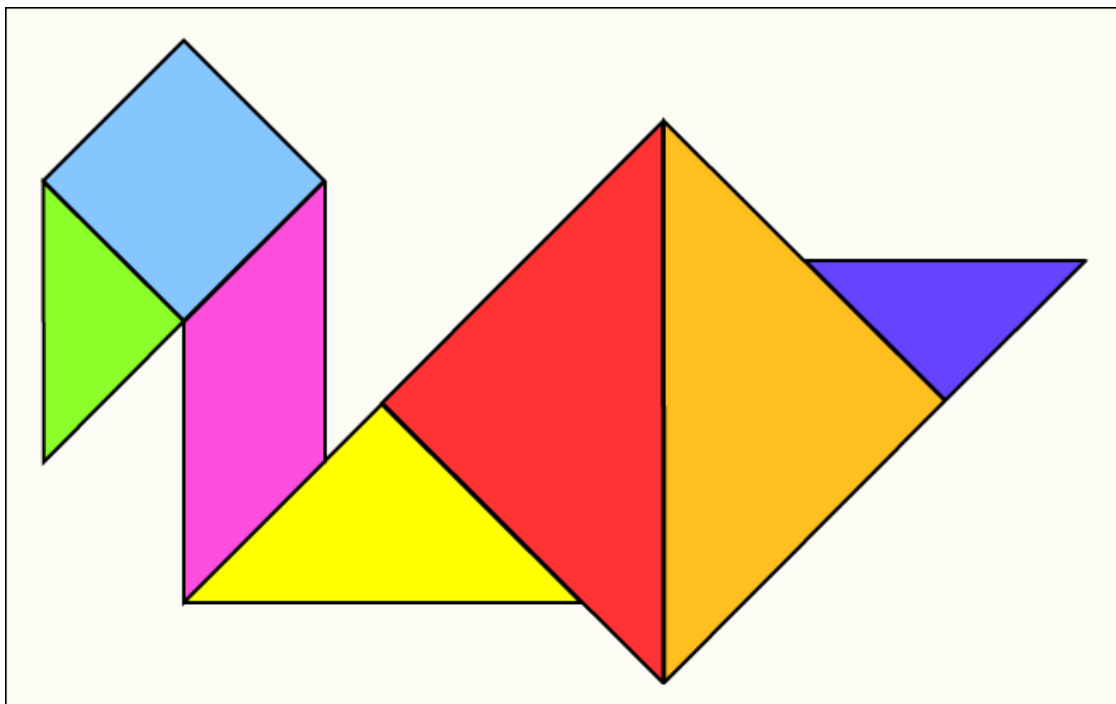
Po teoretičnem uvodu sledi praktični del. Učenci morajo zgraditi določeno obliko iz tangram ploščic, ki so prikazane na sliki 12.



Slika 12: Tangram ploščice.

Potek dela:

- 1 Učence razdelimo v skupine po 4 in vsakemu damo sliko, na kateri je sestavljena določena figura iz tangram ploščic. Primer je prikazan na sliki 13.
  - prvi učenec vidi sliko, ampak daje samo navodila,
  - drugi in tretji učenec ne smeta videti figure, ampak jo gradita glede na navodila, ki jih podaja prvi učenec,
  - četrti učenec šteje, koliko napotkov je bilo danih, preden sta končala figuro, in zapisuje navodila.
- 2 Čas lahko omejimo na 5 minut in po petih minutah mora vsaka skupina pokazati figuro, ki jo je zgradila.
- 3 Četrti učenec pove, koliko navodil je bilo potrebnih za dokončanje modela.



Slika 13: Figura iz tangram ploščic.

Preko tega primera otroci spoznajo, da je algoritem le skupek navodil, ki morajo biti napisana v pravilnem vrstnem redu. Z večkratnimi poskusi spoznajo, da ni nujno, da obstaja vedno samo eno navodilo, vendar pa je od navodil odvisno, v kakšnem času bo problem rešen in koliko korakov bo potrebnih.

## 4 Urejanje z zlivanjem

Pri velikem številu podatkov je pomembno, da so urejeni, saj nam to precej olajša njihovo iskanje. Zato je urejanje v računalništvu in tudi drugod zelo pomembno. Podatki so urejeni v telefonskih imenikih, davčnih registrih, stvarnih kazalnih, knjižnicah, slovarjih, skladiščih, torej povsod, kjer moramo shranjene predmete iskati in dosegati. Že kot majhne otroke so nas učili, da stvari spravljamo "v red". Tako smo se z urejanjem srečali že veliko prej, preden smo se začeli učiti matematiko ali računalništvo.

Kot smo že velikokrat omenili, ima računsko razmišljanje velik vpliv v izobraževanju. Zato bomo v tem poglavju s pomočjo animacije, ki prikazuje urejanje Lego kock glede na njihovo velikost, poskušali predstaviti algoritem urejanja z zlivanjem na čim bolj preprost in razumljiv način. Za začetek si bomo pogledali računalniško definicijo urejanja z zlivanjem in predstavili algoritem.

### 4.1 Opis algoritma

Urejanje z zlivanjem je algoritem za urejanje podatkov v tabeli, in sicer gre za primer algoritma, ki deluje po načelu deli in vladaj. Strategija deli in vladaj deli problem na manjše probleme, ki so prvotnemu problemu enaki (enakega tipa). Postopek ponavljamo, dokler nismo sposobni rešiti podproblemov. Strategija uporablja rekurzijo.

Predpostavimo, da imamo v tabeli  $n$  neurejenih elementov in jih želimo urediti od najmanjšega do največjega. Primer je prikazan na sliki 16. Urejanje poteka v dveh korakih [16]:

1. *Faza delitve*: tabelo razdelimo na dva dela enake velikosti (v splošnem se velikosti lahko razlikujeta za največ en element). Dela, ki smo ju dobili, ponovno razdelimo na dva dela. Delitve ponavljamo s pomočjo rekurzije tako dolgo, dokler v posameznem delu ne dobimo samo enega elementa. Psevdokoda delitve je prikazana na sliki 14.

```

funkcija urejanje_z_zlivanjem (seznam m)
    če dolžina(m) <= 1
        vrni m
    seznam levo, desno
    število sredina = dolžina (m) / 2
    za vsak x iz seznama m ki je manjši od sredina
        dodaj x v seznam levo
    za vsak x v seznamu m ki je večji ali enak sredina
        dodaj x v seznam desno
    levo = urejanje_z_zlivanjem (levo)
    desno = urejanje_z_zlivanjem (desno)
    vrni zlivanje (levo, desno)

```

Slika 14: Pseudokoda delitve tabele pri urejanju z zlivanjem.

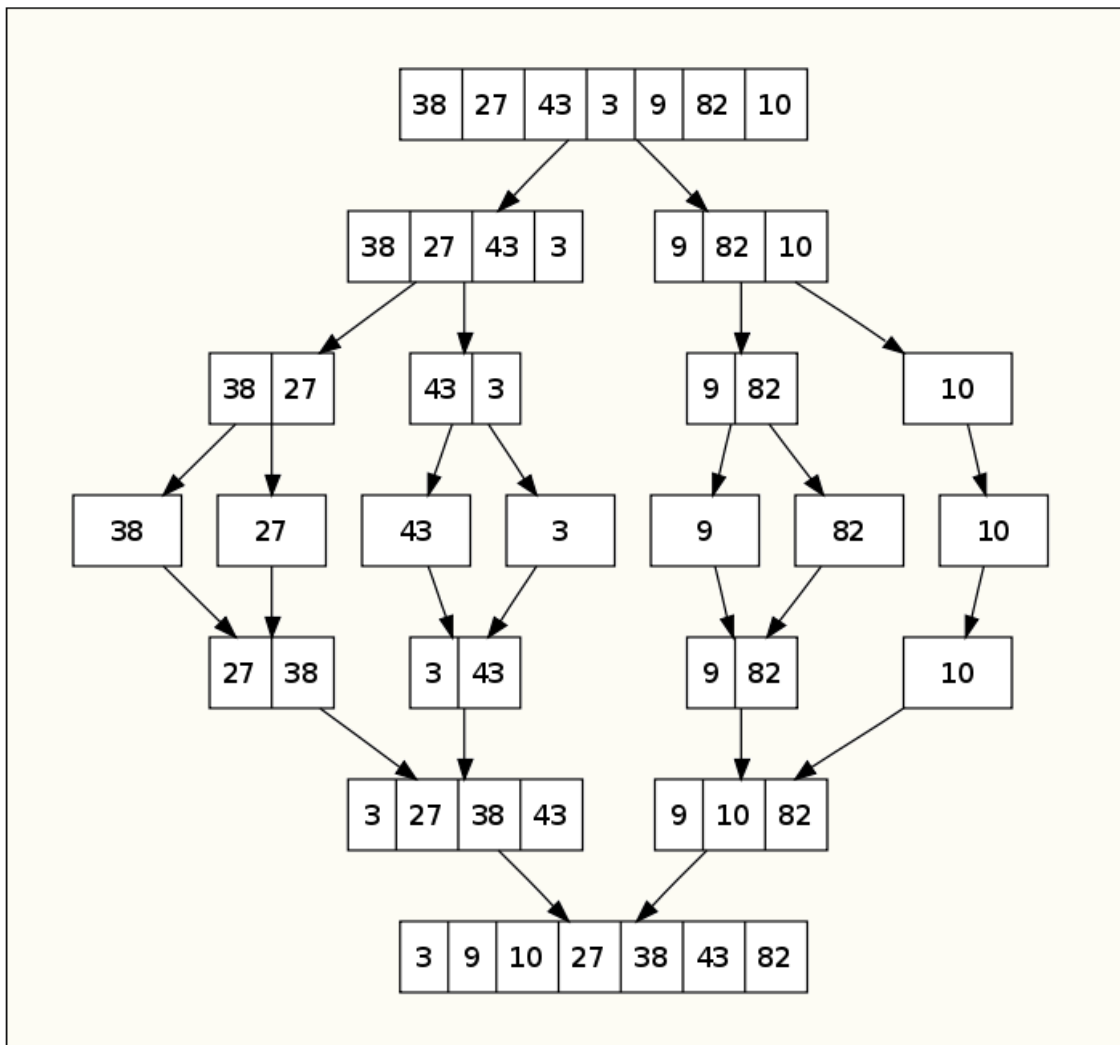
2. *Faza združevanja:* v tej fazi primerjamo dve tabeli med seboj in sicer tako, da iz vsake tabele vzamemo en element, ju primerjamo med seboj in manjšega od njiju zapišemo v novo tabelo. Pseudokoda prikazuje slika 15.

```

funkcija zlivanje (levo, desno)
    seznam urejen_seznam
    dokler dolžina (levo) > 0 ali dolžina (desno) > 0
        če dolžina (levo) > 0 in dolžina (desno) > 0
            če prvi_element (levo) <= prvi_element (desno)
                dodaj prvi_element (levo) v urejen_seznam
                levo = preostanek (levo)
            drugače
                dodaj prvi_element (desno) v urejen_seznam
                desno = preostanek (desno)
        ali če dolžina (levo) > 0
            dodaj prvi_element (levo) v urejen_seznam
            levo = preostanek (levo)
        ali če dolžina (desno) > 0
            dodaj prvi_element (desno) v urejen_seznam
            desno = preostanek (desno)
    vrni končni_seznam

```

Slika 15: Pseudokoda združevanja (zlivanja) pri urejanju z zlivanjem.



Slika 16: Konkreten primer urejanja z zlivanjem na tabeli dolžine 7 [16].

## 4.2 Lastnosti algoritma

### Časovna zahtevnost

Algoritem urejanje z zlivanjem ima dve fazi. Pri računanju časovne zahtevnosti obravnavamo vsako fazo posebej [3].

*Delitev:* V fazi delitve porabimo:

- za izračun sredine tabele konstanten čas:  $\theta(1)$
- za rekurzivno delitev vsake podtabele po strategiji deli in vladaj čas:  $2T(n/2)$

*Združevanje:* za fazo združevanja porabimo čas  $\theta(n)$ .

$$\text{Skupni čas: } T(n) = \begin{cases} \theta(1) & \text{če } n = 1 \\ 2T(n/2) + \theta(n) & \text{če } n > 1 \end{cases}$$

Ker je  $T(n) = \theta(n + n/2 + n/4 + \dots) = \theta(n \cdot \log n)$ , lahko rečemo da za urejanje  $n$  elementov z algoritmom urejanje z zlivanjem, porabimo  $\theta(n \cdot \log n)$  časa.

### Prostorska zahtevnost

Algoritem urejanje z zlivanjem porabi  $\theta(n)$  prostora.

### Primerjava z algoritmom za hitro urejanje

Značilno je, da je algoritem za hitro urejanje (angl. Quicksort) v praksi hitrejši kakor ostali algoritmi, ki imajo prav tako povprečno časovno zahtevnost  $\theta(n \cdot \log n)$ . Vendar pa obstajajo primeri, ko je urejanje z zlivanjem boljša izbira, saj ima hitro urejanje v najslabšem primeru časovno zahtevnost  $\theta(n^2)$ . Urejanje z zlivanjem se največkrat uporablja na povezanih seznamih, saj je v tem primeru razmeroma preprosto implementirati algoritem tako, da potrebuje le  $\theta(1)$  dodatnega prostora. Zlivanje tudi zagotavlja, da ostane vrstni red enakih elementov nespremenjen, kar je pomemben vidik pri urejanju po več kategorijah.

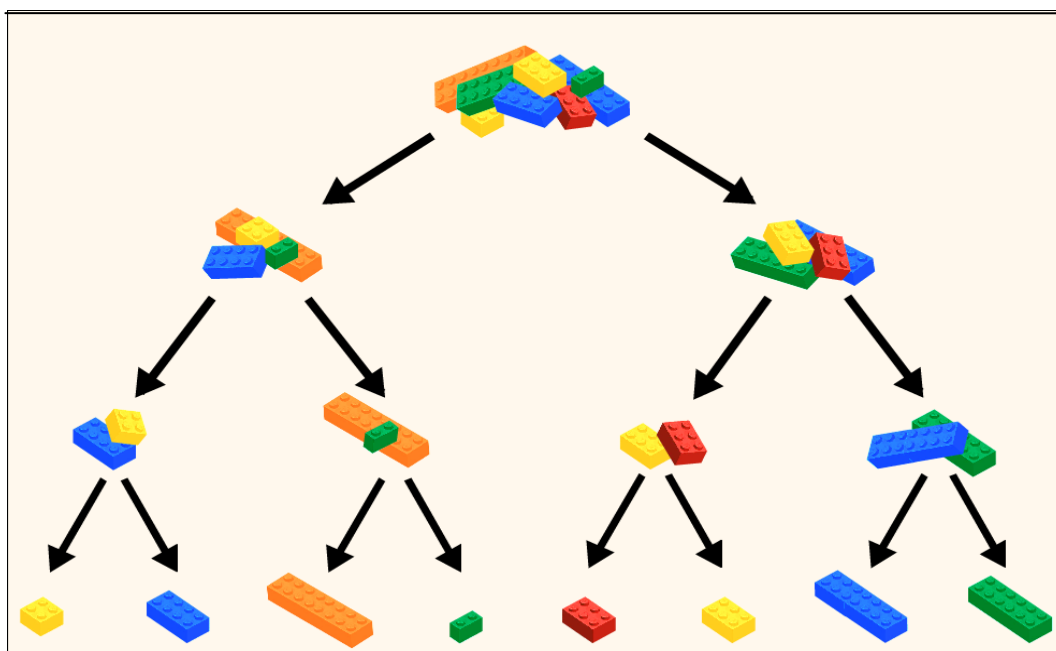
## 4.2 Animacija urejanja Lego kock

### 4.3.1 Orodje Adobe Flash

Pri izdelavi animacije, smo uporabljali program Adobe Flash Professional CS 5.5. Program Flash je namenjen oblikovanju in postavitvi tako enostavnih kot zahtevnejših spletnih strani, ustvarjanju animacij in igrice. Njegova največja prednost je možnost dela z vektorsko grafko, kar nam omogoča hitro in preprosto delo z majhnimi in kakovostnimi datotekami. Animacije so lahko interaktivne (sami odločamo o poteku animacije), vsebujejo lahko glasbo (v zapisu MP3), omogočajo pa tudi programiranje. Program veliko dela opravi kar sam. Na primer pri premikanju objektov določimo samo začetni in končni položaj, vmesno pot pa Flash izračuna sam [10].

### 4.3.2 Animacija v slikah

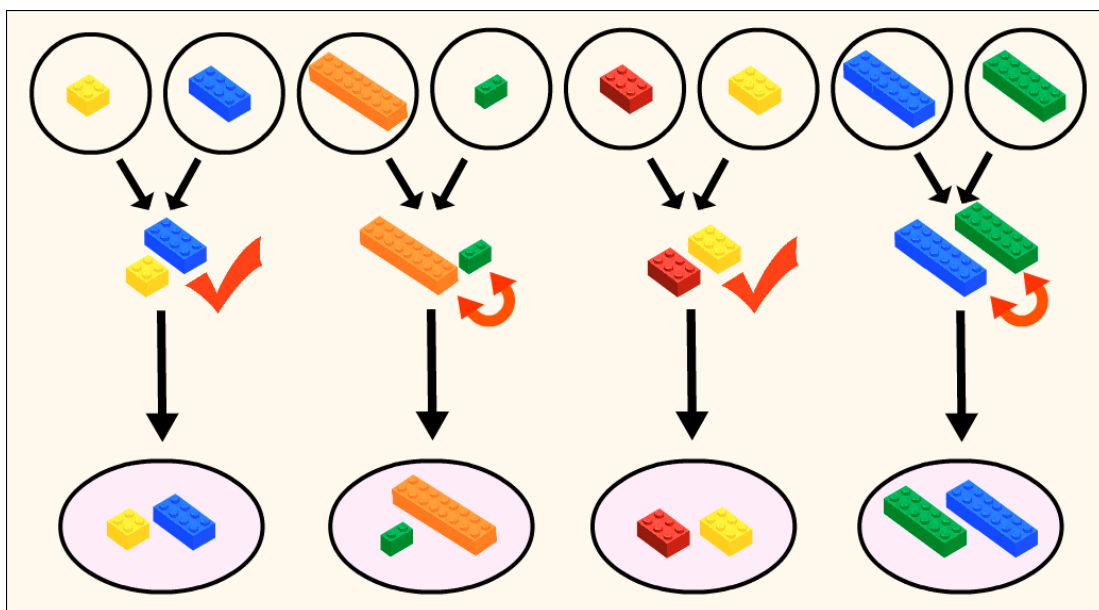
V tem razdelku je prikazana animacija urejanja z zlivanjem na primeru Lego kock. Slika 17 prikazuje prvo fazo algoritma – to je faza delitve. Kocke se rekurzivno delijo na dva dela toliko časa, da na vsakem kupčku ostane le ena kocka.



Slika 17: Faza delitve: rekurzivna delitev kupa Lego kock.

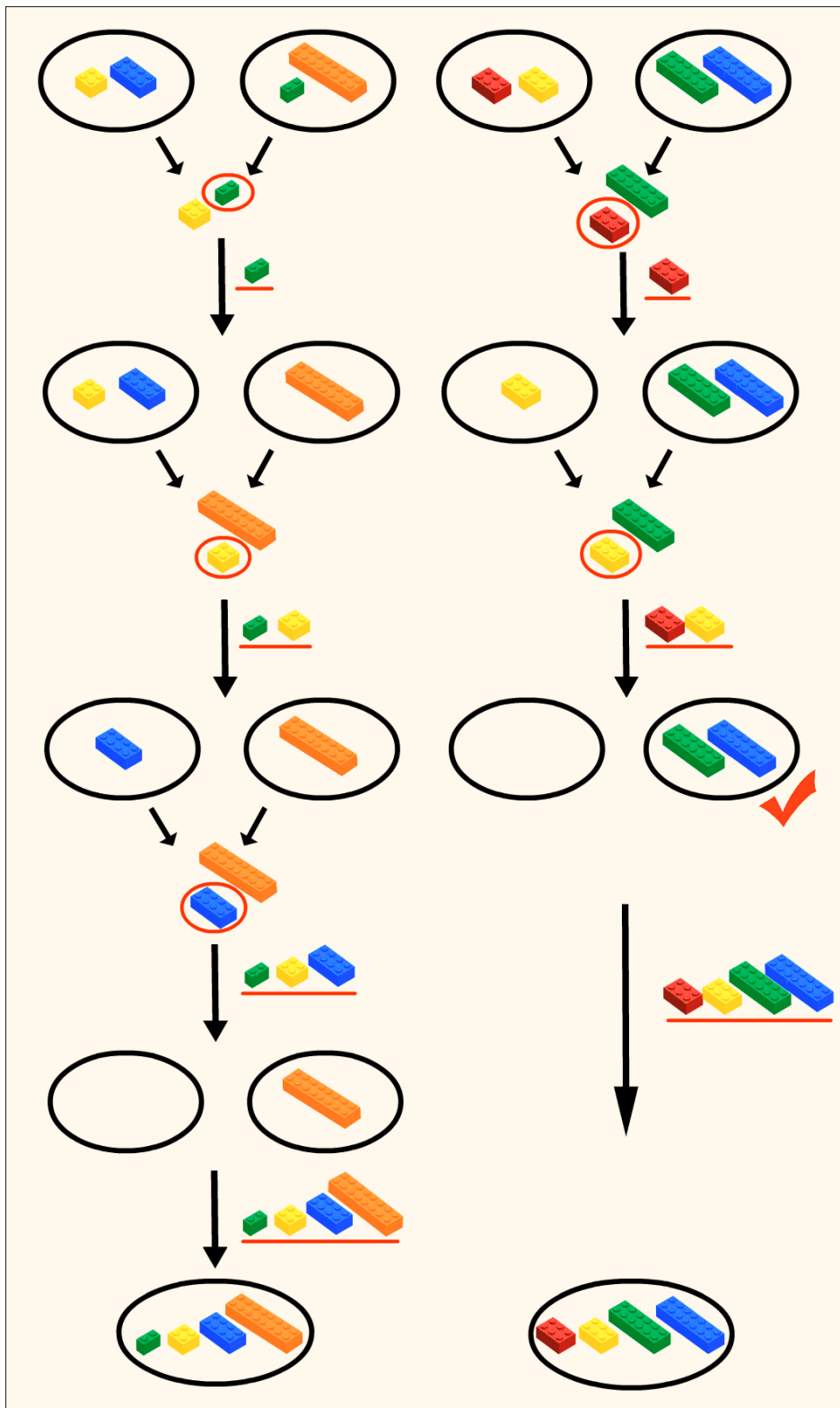


Na sliki 18 je prikazana prva faza združevanja. Dve sosednji kocki primerjamo po velikosti in ju združimo tako, da damo manjšo na prvo mesto novega kupa, večjo pa na drugo. V primeru, ko sta Lego kocki enako veliki, ju samo združimo. To ponovimo za vse Lego kocke. Na koncu dobimo urejen pare kock.



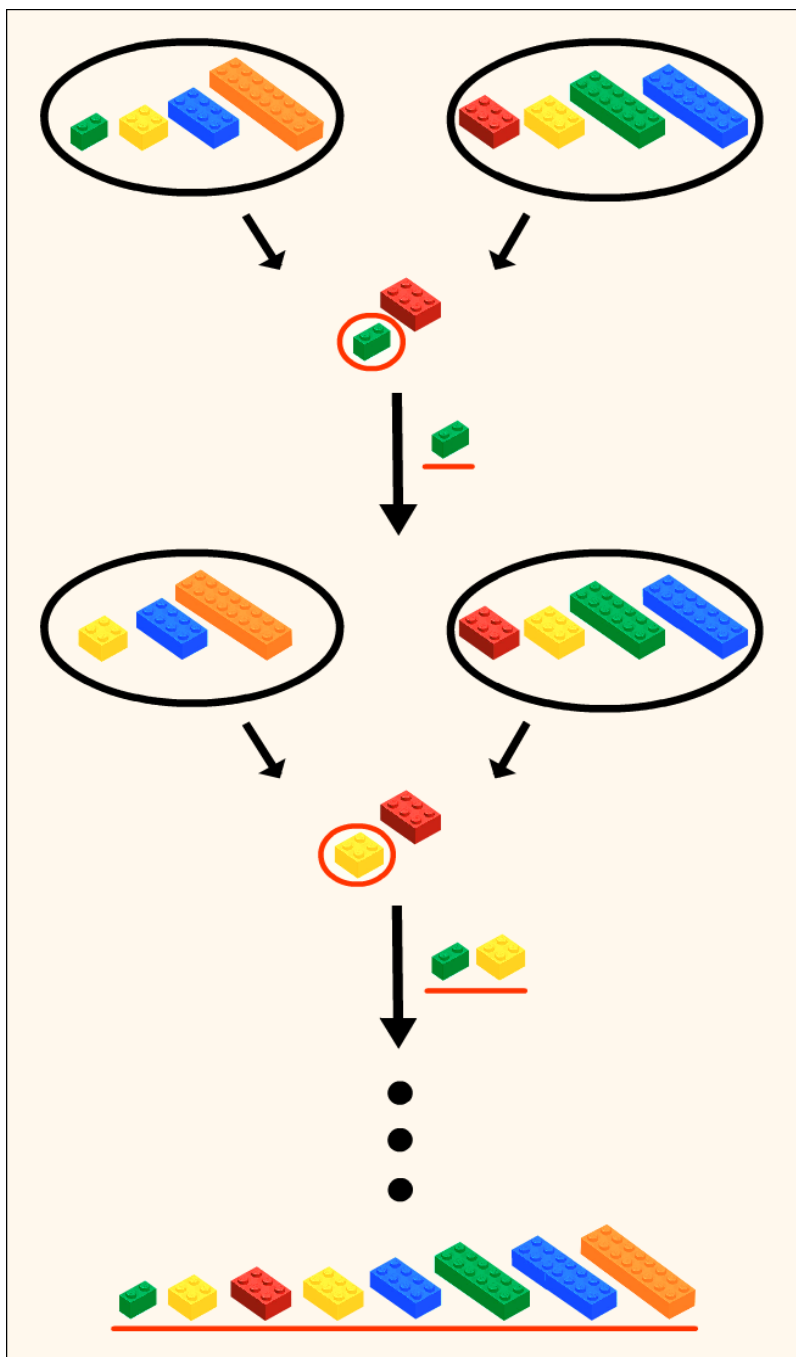
Slika 18: Faza združevanja: primerjamo dve sosednji kocki in manjšo postavimo na prvo mesto, večjo pa na drugo.

Slika 19 prikazuje drugo fazo združevanja – urejanje parov. Ko primerjamo dva para, najprej vzamemo prvo Lego kocko z leve iz vsakega para in ju primerjamo. Manjšo kocko postavimo na nov kup, večjo pa vrnemo na prvotno mesto. V naslednjem koraku spet vzamemo eno kocko iz levega in eno iz desnega kupa in ponavljamo prejšnji korak. V primeru, ko na nekem kupu ni več kock, iz preostalega kupa kocke le prenesemo, saj so že urejene iz prejšnjega koraka. To ponavljamo toliko časa, da dobimo urejene četvorke.



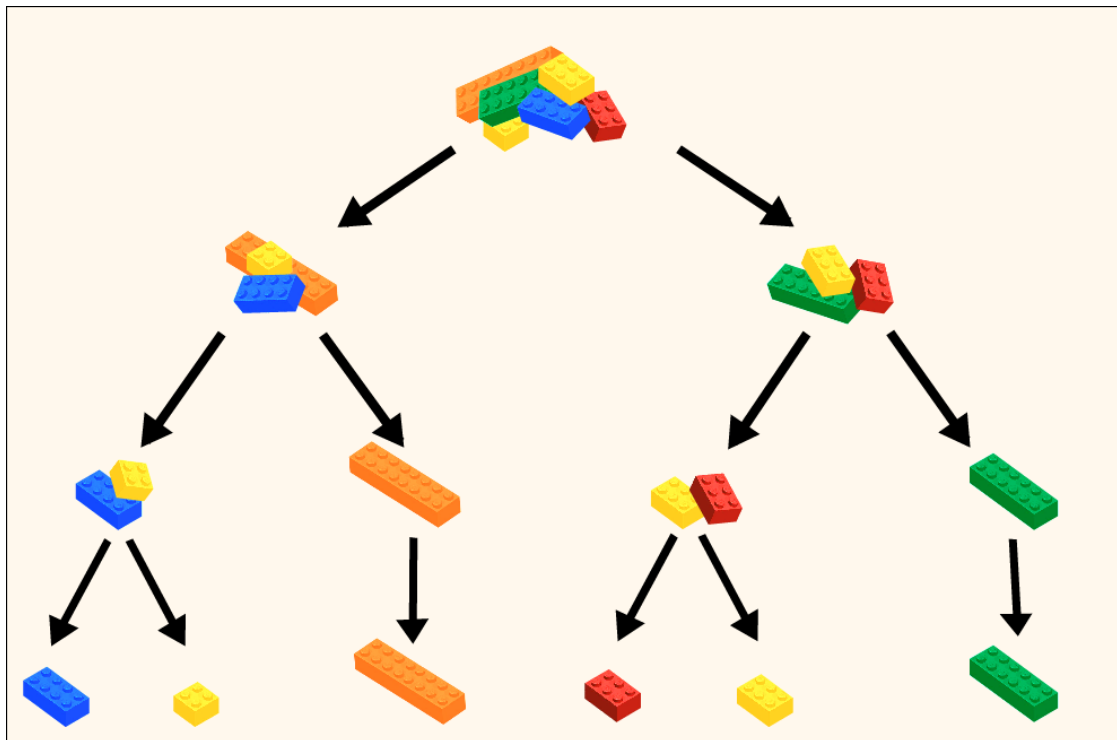
Slika 19: Faza združevanja: primerjanje parov kock.

Slika 20 prikazuje zadnje združevanje, ki poteka po istem principu, kot smo ga opisali v prejšnjem primeru. Na koncu dobimo urejeno vrsto Lego kock po velikosti (od najmanjše do največje).



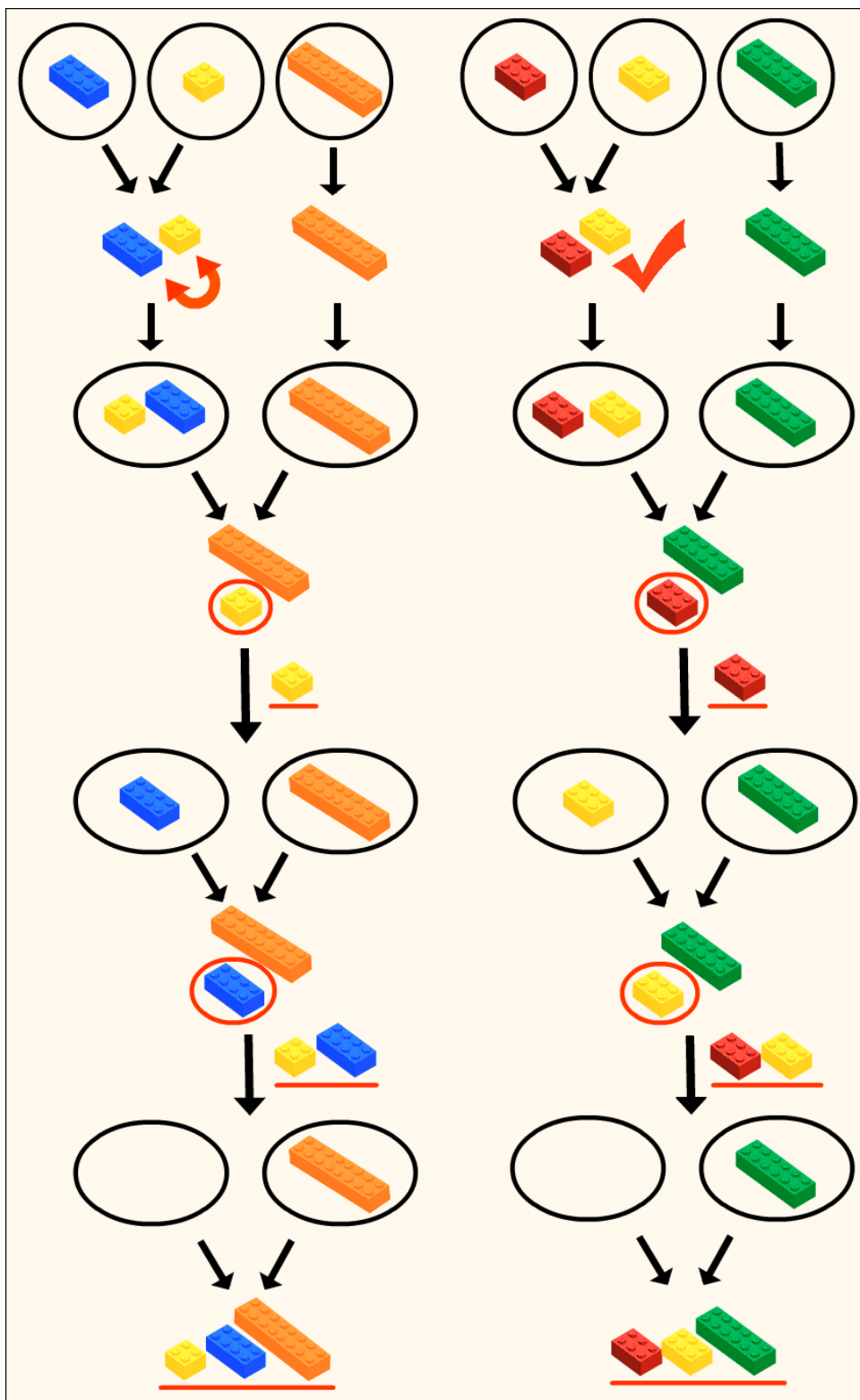
Slika 20: Zadnja faza združevanja in končna rešitev.

Algoritma urejanje z zlivanjem ne uporabljamo le za množice elementov, katerih velikost je potenca števila 2. Učinkovito deluje tudi na preostalih množicah. Pozorni moramo biti le na to, kako delimo in združujemo podmnožice lihih velikosti. V primeru, ko moramo deliti liho število elementov, postavimo v levo podmnožico en element več kot v desno. Primer delitve šestih Lego kock, je prikazan na sliki 21.

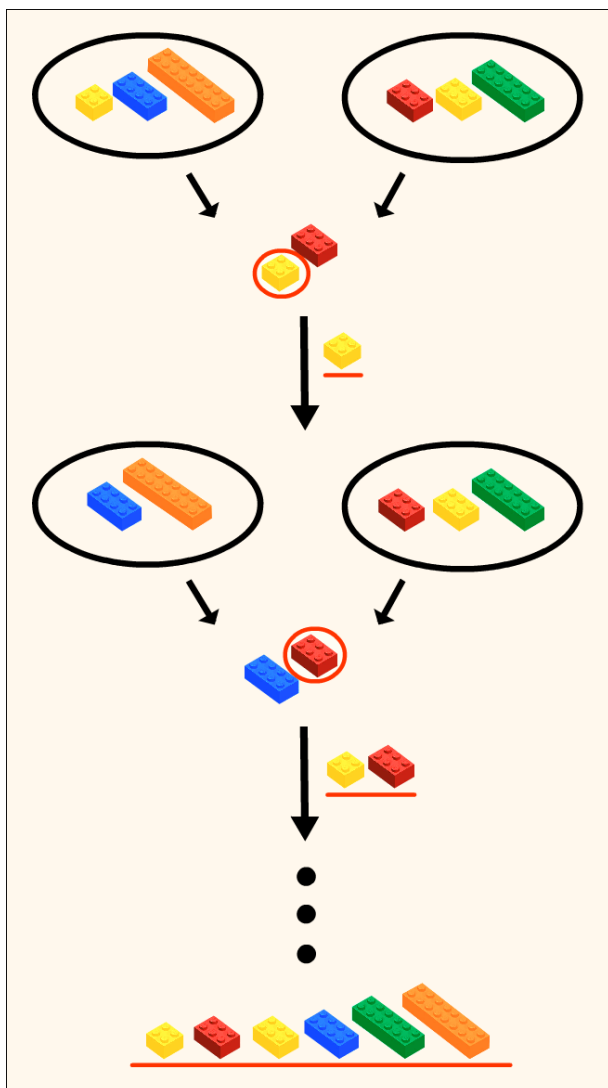


Slika 21: Delitev šestih Lego kock, v kateri pride tudi do delitve lihega števila kock.

V fazi združevanja moramo biti pozorni na to, da združujemo Lego kocke v obratnem vrstnem redu, kakor smo jih delili. To pomeni da najprej združujemo levo podmnožico kock, nato pa še desno, kot to prikazuje slika 22. Zadnja faza združevanja poteka enako kot v prejšnjem primeru, kar je prikazano na sliki 23.



Slika 22: Združevanje leve in desne podmnožice Lego kock.



Slika 23: Zadnja faza združevanja in končna rešitev (6 kock).

Pri animaciji sem se osredotočila le na pravokotne Lego kocke z največjo dolžino 8. Lahko bi urejali kocke različnih velikosti in oblik, s tem da bi na samem začetku določili, katera oblika ima prednost pred drugo. Prav tako bi lahko urejali tudi glede na barvo kock. To bi naredili tako, da bi vnaprej določili barvno lestvico, na kateri bi bilo vidno, katera barva ima prednost.



## 5 Zaključek

Veliko ljudi enači računalništvo s programiranjem. Nekateri starši vidijo le ozek spekter možnosti za zaposlitev svojih otrok, ki se ukvarjajo s področjem računalništva. Mnogi mislijo, da so temeljne raziskave na področju računalništva končane in ostaja le še inženirstvo. Računsko razmišljanje je vizija za usmerjanje učiteljev računalništva, raziskovalcev in vseh, ki želijo spremeniti mnenje družbe o tem. Računalništvo ni samo programiranje. Razmišljati kot računalničar zahteva razmišljanje na različnih ravneh abstrakcije. Prav tako pri računalništvu nista pomembni le programska in strojna oprema, ampak tudi koncepti, ki jih lahko uporabljamo v vsakdanjem življenju.

Računsko razmišljanje je način, kako ljudje rešujemo probleme, in nas skuša pripraviti, da bi delovali kot računalniki. Omogoča nam, da računanje prilagodimo našim potrebam. Eden izmed temeljev računskega razmišljanja je logično razmišljanje. Pri logičnem razmišljanju s pomočjo že znanih dejstev sklepamo na nove informacije. Iskanje, načrtovanje in razporejanje so vidiki računalniškega razmišljanja, ki nas spremljajo v vsakdanjem življenju.

Ker je uporaba računalniških naprav v vsakdanjem življenju postala zelo razširjena, včasih pravimo, da živimo v obdobju digitalnih informacij ali informacijske revolucije. To je še posebej očitno na področju izobraževanja na vseh ravneh. Veliko truda bo še potrebno vložiti, da bomo učence v osnovnih in srednjih šolah seznanili z vsaj osnovnimi in bistvenimi računskimi koncepti.

V diplomski nalogi smo predstavili metode računskega razmišljanja, predvsem pa smo se osredotočili na pomen računskega razmišljanja v raznih disciplinah, kot so biologija, kemija, matematika, novinarstvo in medicina. Ker se sama navdušujem nad pedagoškim delom na področju računalništva in matematike, sem največ pozornosti namenila pomenu računskega razmišljanja v izobraževanju. Pri pisanju diplome sem spoznala veliko načinov, kako otrokom približati in na zanimiv način razložiti osnovne koncepte in metode tako iz področja matematike kot tudi računalništva. Izdelala sem primer urejanja z zlivanjem in pripravila animacijo namenjeno otrokom. Animacija temelji na urejanju Lego kock po velikosti od najmanjše do največje. Pri izdelavi animacije sem se podrobneje seznanila s programom Flash, ki je enostaven za tovrstno delo.



Ideja računskega razmišljanja se mi zdi zelo zanimiva in koristna, še posebej na področju izobraževanja. Poznavanje računskega razmišljanja bi marsikateremu učitelju lahko pomagalo pri izboljšanju učnih ur. Menim, da bi bilo potrebno trenutni slovenski učni načrt računalništva in informatike, ki temelji na poučevanju in uporabi pisarniških aplikacij in nekaterih programskih jezikov, preoblikovati tako, da bi se otroci učili osnovnih konceptov računalništva, kot so algoritmi, rekurzija, sklad, itd.

# Seznam slik

Slika 1: Program za izračun razlike (v odstotkih) v jeziku Python .....	14
Slika 2: Razrez DNK na različno dolge fragmente.....	26
Slika 3: Iskanje fragmenta dolžine 2000 baznih parov.....	26
Slika 4: Kloniranje fragmenta s pomočjo bakterije .....	27
Slika 5: Fragment z 500 baznimi pari nukleotidov .....	27
Slika 6: Iskanje prekrivajočih se koncev fragmentov in združitvev v eno verigo.....	28
Slika 7: RoboZZle - ukazi za premikanje robota po igralni plošči.....	30
Slika 8: RoboZZle - ukazi za klic funkcije .....	31
Slika 9: RoboZZle - pogojni ukaz.....	31
Slika 10: RoboZZle - Rekurzivne funkcije.....	32
Slika 11: RoboZZle - Izvajanje rekurzivne funkcije z uporabo sklada .....	33
Slika 12: Tangram ploščice.....	35
Slika 13: Figura iz tangram ploščic .....	36
Slika 14: Psevdokoda delitve tabele pri urejanju z zlivanjem. ....	38
Slika 15: Psevdokoda združevanja (zlivanja) pri urejanju z zlivanjem. ....	38
Slika 16: Konkreten primer urejanja z zlivanjem na tabeli dolžine.....	39
Slika 17: Faza delitve: rekurzivna delitev kupa Lego kock.....	41
Slika 18: Faza združevanja: primerjava dveh sosednjih kock .....	42
Slika 19: Faza združevanja: primerjanje parov kock.....	43
Slika 20: Zadnja faza združevanja in končna rešitev.....	44
Slika 21: Delitev šestih Lego kock .....	45
Slika 22: Združevanje leve in desne podmnožice Lego kock.....	46
Slika 23: Zadnja faza združevanja in končna rešitev (6 kock). ....	47



# Seznam tabel

Tabela 1: Discipline, ki jih združujejo na univerzi Carnegie Mellon. ....	18
Tabela 2: Uporaba računalniških konceptov na različnih področjih .....	22



# Literatura

- [6] V. Barr, C. Stephenson, Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community?, *ACM Inroads*, 2011, str. 48-54.
- [2] A. Blum, T. Sandholm, D. J. Abraham, Clearing Algorithms for Barter Exchange Markets: Enabling Nationwide Kidney Exchanges, *Proceedings of the ACM Conference on Electronic Commerce (EC)*, San Diego, California, 2007.
- [3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to algorithms, 3rd Edition, Cambridge: The MIT Press, 2009, str. 34-37.
- [4] S. Papert, An Exploration in the Space of Mathematics Educations, *International Journal of Computers for Mathematical Learning*, 1996, str. 95-123.
- [5] Pearson K., Why journalists need computational thinking, [Splet]. Dostopno na: <http://kimpearson.net/?p=61>. (Datum dostopa: 4. 2. 2012).
- [6] J. M. Wing, Computational Thinking, *Communications of the ACM*, Vol. 49, No. 3, marec 2006, str. 33-35.
- [7] J. M. Wing, Research Notebook: Computational Thinking - What and Why?, *The Link: News from the School of Computer Science*, 2011, str. 20-23.
- [8] Center for Computational Thinking, Carnegie Mellon, Dostopno na: <http://www.cs.cmu.edu/~CompThink/seminars.html#>. (Datum dostopa: 20. 2. 2012).
- [9] Computational chemistry, Dostopno na: [http://en.wikipedia.org/wiki/Computational\\_chemistry](http://en.wikipedia.org/wiki/Computational_chemistry). (Datum dostopa: 15. 1. 2012).
- [10] Flash vodič, Dostopno na: <http://moje-mesto.com/adobe-flash-vodici>. (Datum dostopa: 6. 3. 2012).
- [11] Fun and Games with Computatinal Thinking: Children Thinking Logically, Dostopno na: <http://cloudparenting.org/2011/07/08/fun-games-computational-thinking-thinking-logically/>. (Datum dostopa: 3. 1. 2012).

- [12] Google: Exploring Computational Thinking,“ Dostopno na:  
<http://www.google.com/edu/computational-thinking/what-is-ct.html>. (Datum dostopa: 10. 2. 2012).
- [13] humanShot.swf (Predmet application/x-shockwave-flash), Dostopno na:  
<http://smcg.ccg.unam.mx/enp-unam/03-EstructuraDelGenoma/animaciones/humanShot.swf>. (Datum dostopa: 7. 3. 2012).
- [14] ISTE|CT Toolkit, Dostopno na:  
[http://www.iste.org/learn/computational-thinking/computational-thinking\\_toolkit.aspx](http://www.iste.org/learn/computational-thinking/computational-thinking_toolkit.aspx). (Datum dostopa: 11. 1. 2012).
- [15] Kaj je robotska kirurgija, Dostopno na:  
[http://en.wikipedia.org/wiki/Robotic\\_surgery](http://en.wikipedia.org/wiki/Robotic_surgery). (Datum dostopa: 12. 2. 2012).
- [16] Merge sort, Dostopno na:  
[http://en.wikipedia.org/wiki/Merge\\_sort](http://en.wikipedia.org/wiki/Merge_sort). (Datum dostopa: 5. 3. 2012).
- [17] Previous Seminars, Dostopno na:  
<http://www.inf.ed.ac.uk/research/programmes/comp-think/previous.html>. (Datum dostopa: 12. 2. 2012).
- [18] Robotic surgery, Dostopno na: [http://en.wikipedia.org/wiki/Robotic\\_surgery](http://en.wikipedia.org/wiki/Robotic_surgery).  
(Datum dostopa: 12. 2. 2012).
- [19] RoboZZle - RoboZZle Wiki, Dostopno na:  
<http://robozzle.com/wiki/>. (Datum dostopa: 2. 3. 2012).
- [20] Scratch (programming language), Dostopno na:  
[http://en.wikipedia.org/wiki/Scratch\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/Scratch_%28programming_language%29). (Datum dostopa: 20. 2. 2012).