ZAPOREDJE ŽARNIC

Raziskovalna naloga iz področja matematike

Avtorji:

- Tomo Joksimović Terpin, dijak drugega letnika
- Maks Tomšič, dijak drugega letnika

Mentor: mag. Andreja Kramar, profesorica matematike

Somentor: Nastja Lasič, profesorica informatike

Šolsko leto 2022-23

Gimnazija Šentvid

Prušnikova 98, Ljubljana Šentvid

Marec 2023

KAZALO

KLJUČNE BESEDE: periodična zaporedja, seznam sprememb, začetno in končno stanje.

POVZETEK NALOGE:
V raziskovalni nalogi smo opazovali zaporedja neodvisnih žarnic, ki smo jim na urejen način spreminjali stanja. Dopustili smo le take transformacije T_n, kjer se spremeni stanje na vsaki n. žarnici. Opazovali smo, kakšen učinek ima zaporedna izvedba več takih transformacij. Odkrili smo povezavo med začetnim in končnim stanjem, pa tudi algoritem, ki periodičnemu končnemu stanju določi seznam izvedenih transformacij pri pogoju, da so bile na začetku vse žarnice ugasnjene. Na koncu smo napisali še tri programe v programu Python, s katerimi smo preizkusili ugotovitve iz teoretičnega dela.

KEY WORDS: periodic sequences, list of changes, initial and final state.

TASK SUMMARY:
In the research task, we observed sequences of independent light bulbs, whose states were changed in an orderly manner. We only allowed such transformations T_n, where the state changes at each n. light bulb. We observed the effect of successive execution of several such transformations. We have discovered a connection between the initial and final states, as well as an algorithm that determines the periodic final state with a list of performed transformations under the condition that at the beginning all the light bulbs were turned off. In the end, we wrote three more programs in Python, with which we tested the findings from the theoretical part.

1. UVOD – OPREDELITEV NALOGE

Imejmo neskončno zaporedje neodvisnih žarnic, ki so bodisi prižgane bodisi ugasnjene. Zanima nas, kaj se zgodi, če jih na urejen način vklapljamo oz. izklapljamo, to pomeni, da spremenimo stanje na vsaki k. žarnici. Če to izvedemo večkrat, nastanejo zanimiva zaporedja, ne glede na začetno stanje žarnic. Radi bi razumeli taka zaporedja in jih zapisali na kar se da enostaven način.

Hipoteza: Pri znanem začetnem stanju žarnic in pri znanih večkratnih spremembah se da izračunat, ali n. žarnica gori. Zaporedja, ki nastanejo, se da lepo zapisati.

Cilji: Najprej bomo definirali osnovne izraze, ki so povezani z vsebino raziskovalne naloge. Definirali bomo načine, kako se spreminja stanje žarnic. Raziskali bomo ali vrstni red vpliva na končno stanje in kaj se zgodi, če isto spremembo izvedemo večkrat. Poskušali bomo uporabiti učinkovite in smiselne matematične zapise. Ugotovili bomo, kako se izračuna končno stanje žarnic in pogledali nekaj zanimivih zaporedij. Radi bi raziskali, ali se iz končnega stanja da ugotoviti, katere spremembe so bile izvedene. Izdelali bomo simulacijo v programskem jeziku Python.

2. TEORETIČNI DEL

2.1. ZAČETNO STANJE IN SPREMEMBA STANJA

Imejmo neskončno žarnic, ki si sledijo ena za drugo in so prižgane neodvisno druga od druge. Njihovo stanje lahko zapišemo na različne načine:

- S = 0010110110010 … zapis v obliki zlepka ničel in enk,
- S = 0,0,1,0,1,1,0,1,1,0,0,1,0, … zapis v obliki neskončnega zaporedja,
- S = (0,0,1,0,1,1,0,1,1,0,0,1,0, …) zapis v obliki vektorja z neskončno komponentami.

Pri teh zapisih 1 pomeni, da žarnica sveti oziroma 0, da žarnica ne sveti. Med zgornjimi načini smo se odločili za zadnji zapis, saj nam omogoča razmišljanje po komponentah, to pomeni glede na posamezne žarnice. Tu je prikazano zgornje stanje žarnic in vrstni red žarnic.

V nadaljevanju se bomo ukvarjali s spremembami stanj in tudi tu potrebujemo matematičen zapis. **Začetno stanje** označimo z S_0, spremenjena stanja pa potem z S_1, S_2, S_3, … tako bomo dobili zaporedje stanj. Zapis stanj:

S_n = n-to zaporedno stanje
S_0 = začetno stanje žarnic
S_k = končno stanje

Primer 1:
Naj bodo na začetku vse žarnice izklopljene, potem pa zaporedoma izvedemo nekaj sprememb.

Začetno stanje	$S_0 = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, \dots)$
Spremenimo stanje vsake druge žarnice	$S_1 = (0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1, \dots)$
Spremenimo stanje vsake tretje žarnice	$S_2 = (0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,0,0, \dots)$
Spremenimo stanje vsake pete žarnice	$S_3 = (0,1,1,1,1,0,0,1,1,0,0,0,0,1,0,1,0,0, \dots)$

Spremembo stanja ali transformacijo stanja, kjer spremenimo vsako n. žarnico, bomo označili s simbolom T_n. V zgornjem primeru bi spreminjanje lahko zapisali takole:

$$S_0 \to T_2(S_0) \to T_3(S_1) \to T_5(S_2) = S_3$$

Spremembe stanj:

T_n = sprememba vsake n-te žarnice
T_0 = nima učinka, vse ostane enako
T_1 = sprememba pri vseh žarnicah
T_2 = sprememba pri vsaki drugi žarnici
T_3 = sprememba pri vsaki tretji žarnici

Tudi transformacije lahko zapišemo v obliki vektorjev:

$$T_2 = (0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,\dots)$$
$$T_3 = (0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,\dots)$$

Ti zapisi imajo povsod 0, enice se pojavijo le na tistih mestih, ki so deljiva z n.
Transformacija, ki spremeni stanja vseh žarnic je T_1.

$$T_1 = (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,\dots)$$

Transformacija, ki ne spremeni stanja, je T_0.

$$T_0 = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\dots)$$

V nadaljevanju bomo poskušali ugotoviti, kako bi s temi zapisi izračunali stanja žarnic.

2.2. ZAPOREDNE SPREMEMBE STANJ

Zanima nas, ali vrstni red zaporednih sprememb vpliva na končno stanje žarnic.
Vzemimo podatke iz primera 1 in zamenjajmo vrstni red, na primer takole:

$$S_0 \xrightarrow{T_3} S_1 \xrightarrow{T_5} S_2 \xrightarrow{T_2} S_3$$

Zaporedje sprememb je sedaj $(T_3, \quad T_5, \quad T_2)$.

Začetno stanje	$S_0 = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\dots)$
Spremenimo stanje vsake tretje žarnice	$S_1 = (0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,\dots)$
Spremenimo stanje vsake pete žarnice	$S_2 = (0,0,1,0,1,1,0,0,1,1,0,1,0,0,0,0,0,1,\dots)$
Spremenimo stanje vsake druge žarnice	$S_3 = (0,1,1,1,1,0,0,1,1,0,0,0,0,1,0,1,0,0,\dots)$

Opazimo, da je končno stanje enako, kot prej. Je to slučajno?

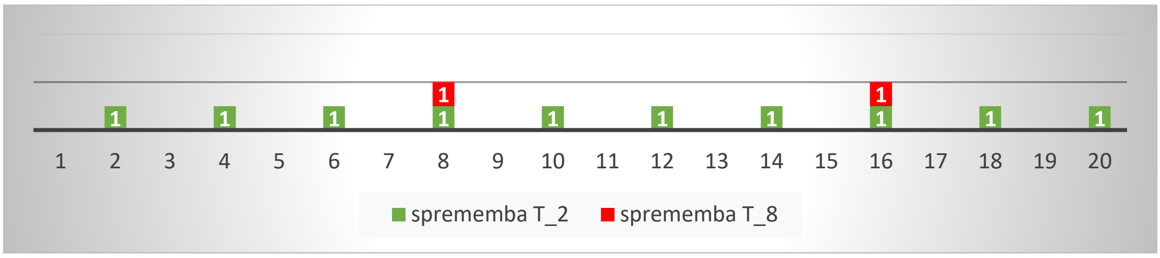
Premislimo, kaj se zgodi, če zamenjamo dve različni transformaciji. Če najprej spremenimo stanje na vsaki n. žarnici, potem pa na vsaki m. žarnici, bodo spremembe na vseh žarnicah, ki so pri večkratnikih števila n ali pa pri večkratnikih števila m, razen, ko se večkratniki ujamejo in se stanje žarnice vrne v prvotno. Pri različnem vrstnem redu seveda ne bo nič drugače. Tudi ni pomembno, če sta števili n in m tuji. Torej velja:

$$(T_n, T_m) = (T_m, T_n)$$

Ugotovili smo, da pri dveh zaporednih različnih spremembah velja:

- stanje žarnice se spremeni, če je njeno zaporedno mesto deljivo bodisi z n bodisi z m, a ne z obema hkrati,
- stanje žarnice se ohrani, če njeno zaporedno mesto ni deljivo niti z n niti z m, ali pa je deljivo z obema hkrati.

Ali velja isto, če je eno število večkratnik drugega? Recimo, da najprej spremenimo stanje vsake 2. žarnice, potem pa še vsake 8. žarnice. Stanje se bo spremenilo pri žarnicah 2,4,6,10,12,14,18,20, … Enako bi bilo, če bi zamenjali vrstni red sprememb.



Kaj pa, če isto transformacijo izvedemo dvakrat zaporedoma? Najprej se stanja žarnic na večkratnikih spremenijo in ob ponovni izvedbi še enkrat. Torej ni nobenega učinka, lahko pa rečemo, da ponovna transformacija izniči učinek prve. Splošno bi zapisali:

$$(T_n, T_n) = T_0$$

Če bi neko transformacijo izvedli večkrat zaporedoma, lahko zanemarimo vse podvojitve. Če gre za sodo število transformacij, se torej izničijo, če pa jih je liho, je učinek isti, kot bi bila le ena.

$$(T_n, T_n, T_n, T_n) = T_0$$

$$(T_n, T_n, T_n, T_n, T_n) = T_n$$

Posledica vseh ugotovitev je, da lahko zaporedne spremembe uredimo po velikosti in izločimo vse podvojitve. Tako dobimo enakovredno skrčitev zaporedja transformacij, ki ji bomo rekli na kratko **seznam sprememb**. Ker vrstni red ni pomemben, ga bomo pisali v obliki množice.

Primer 2:
Skrčitev in seznam sprememb.
$(T_3, \quad T_5, \quad T_2, \quad T_2, \quad T_7, \quad T_5, \quad T_7, \quad T_2, \quad T_7) \;=\; (T_2, \quad T_3, \quad T_7) = \{2,3,7\}$

Zdaj imamo oznake in zapise za spremembe, v nadaljevanju pa bomo z njimi računali. Kot smo ugotovili doslej, nas bodo zanimali večkratniki izbranih števil, ti imajo pri deljenju z njimi ostanek 0. Z ostanki se ukvarjajo kongruence.

2.3. RAČUNANJE PO MODULU

Če imata dve naravni števili a in b enak ostanek pri deljenju z 2, rečemo, da sta **kongruentni po modulu 2** in zapišemo takole:

$$a \equiv b(mod\ 2)$$

Primer 3:
$12 \equiv 10(mod\ 2),\ 3 \equiv 1(mod\ 2)$

Podobno kot zgoraj, opišemo kongruentnost po modulu m. Naravni števili a in b sta **kongruentni po modulu $\boldsymbol{m}$** , če imata enak ostanek pri deljenju z m. Zapis:

$$a \equiv b(mod\ m)$$

Primer 4:
$7 \equiv 2(mod\ 5),\ \ 10 = 1(mod\ 3)$

Z ostanki oziroma kongruencami lahko računamo podobno kot s števili, le da na koncu zapišemo ostanek pri deljenju rezultata po izbranem modulu.

Primer 5:
Prikaz seštevanja, odštevanja, množenja naravnih števil.

$$3 + 4 = 7 \equiv 2(mod5)$$
$$15 - 4 = 11 \equiv 2(mod3)$$
$$3 \cdot 4 = 12 \equiv 0(mod2)$$

Primer 6:
Število a naj ima pri deljenju s 3 ostanek 2, število b prav tako. Njuna vsota ima pri deljenju s 3 ostanek 1.

$$a \equiv 2(mod\ 3),\ \ b \equiv 2(mod\ 3)$$
$$a + b = 4 \equiv 1(mod\ 3)$$
$$a + b \equiv 1(mod\ 3)$$

Tu smo napisali samo osnovna dejstva, ki jih bomo potrebovali v nadaljevanju. Ker sta pri žarnicah možni le dve stanji, nas bodo zanimali ostanki pri deljenju z 2. Možna sta le dva ostanka, to sta 0 in 1, zato velja

- za soda števila: $n \equiv 0(mod2)$
- za liha števila: $n \equiv 1(mod2)$

To pravilo lahko napišemo tudi v obliki funkcije z dvema možnima izhodoma:

$$f(n) = \begin{cases} 0; & n\ sodo \\ 1; & n\ liho \end{cases} = \begin{cases} 0; & n \equiv 0(mod\ 2) \\ 1; & n \equiv 1(mod\ 2) \end{cases}$$

V naši nalogi se ukvarjamo s transformacijami stanj in jih lahko sedaj napišemo tudi v obliki funkcije. Transformacija T_n spremeni stanje vsake n-te komponente. Na k-tem mestu bo 1, če je k deljivo z n, drugače pa bo 0. **Vsaka transformacija deluje (učinkuje) od n. žarnice naprej.**

$$T_n = (0,0,0\ldots,1,0,0,0\ldots,1,0,0,0\ldots)$$

Zapis v obliki funkcije ima dve spremenljivki, k nam pove vrstni red žarnice.

$$T_n = f(n,k) = \begin{cases} 1; & k \equiv 0(mod\ n) \\ 0; & v\ drugih\ primerih \end{cases}$$

V prihodnje ne bomo računali s tem zapisom, lahko pa bi bil začetek kake druge raziskovalne naloge.

2.4. RAČUNANJE PO KOMPONENTAH

Računanje po komponentah smo spoznali pri vektorskem računu. To pomeni, da vzporedno izvajamo operacije na vsaki komponenti posebej, kot je prikazano v naslednjem primeru.

Primer 7:
Štiri različne račune lahko zapišemo ločeno (levo) ali pa združeno v komponentah (desno).

$$\begin{array}{l} 1+3=4 \\ 2+5=7 \\ 1+2=3 \\ 1+3=4 \end{array} \qquad\qquad \begin{array}{r} (1,2,1,3) \\ +(3,5,2,1) \\ \hline (4,7,3,4) \end{array}$$

Računanje po komponentah je zelo pregledno in hitro. Tak način bi lahko uporabili pri računanju stanja žarnic. Hitro smo ugotovili (malo po občutku), da lahko spremembo stanja izračunamo s prištevanjem vektorja, ki predstavlja transformacijo.

Primer 8:
Vse žarnice so najprej ugasnjene, nato spremenimo vsako tretjo in potem še vse.

$$\begin{array}{rl} S_0 = & (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\ldots) \\ +T_3 = & (0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,\ldots) \\ +T_1 = & (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,\ldots) \\ \hline S_1 = & (1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,\ldots) \end{array}$$

Podobno računamo, če je začetno stanje poljubno. Tedaj bi pri vsoti lahko dobili tudi 2, a to pomeni, da se žarnici stanje ne bo spremenilo. Zapisati moramo le ostanek pri deljenju z 2, to je vsota po modulu 2.

$$\begin{array}{rl} S_0 = & (1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\ldots) \\ +T_3 = & (0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,\ldots) \\ \hline S_1 = & (1,0,2,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,\ldots) \\ S_1 = & (1,0,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,\ldots) \end{array}$$

Pri dveh ali več zaporednih spremembah potem na podoben način prištejemo vektorje sprememb in na koncu ostanke pri deljenju z 2.

Primer 9:
Tri spremembe in končno stanje.

$$\begin{aligned} S_0 &= (1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,\dots) \\ +T_2 &= (0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,\dots) \\ +T_3 &= (0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,\dots) \\ +T_5 &= (0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,\dots) \\ \hline S_3 &= (1,1,1,1,1,\mathbf{3},0,1,1,\mathbf{2},0,\mathbf{2},0,1,\mathbf{2},1,0,\mathbf{2},\dots) \\ S_3 &= (1,1,1,1,1,\mathbf{1},0,1,1,\mathbf{0},0,\mathbf{0},0,1,\mathbf{0},1,0,\mathbf{0},\dots) \end{aligned}$$

Tukaj je pomembno, da ne menjamo vrstnega reda komponent. Modul lahko upoštevamo sproti, najlažje pa je na koncu. Zdaj lahko tudi z računi potrdimo ugotovitve o zaporednih spremembah stanj iz poglavja 3.2.

2.5. RAČUNANJE KONČNEGA STANJA

Recimo, da poznamo začetno stanje žarnic S_0 in skrčen seznam sprememb, kjer so števila po velikosti in se ne podvajajo. **Končno stanje dobimo tako, da začetnemu stanju prištejemo vse transformacije T_n po modulu 2.**

$$S_k = (S_0 + T_n)(mod\ 2)$$

$$S_k = (S_0 + T_a + T_b + T_c + \cdots)(mod\ 2)$$

Premislimo, če lahko končno stanje opišemo še drugače. Stanje žarnice na m. mestu se bo spremenilo, če je m večkratnik lihega števila števil iz seznama sprememb. Če je število m večkratnik sodega števila števil iz seznama, pa do spremembe ne bo prišlo.

Splošna ugotovitev: Stanje na m. žarnici se ohrani, če je število m deljivo s sodim številom števil iz seznama sprememb. V nasprotnem primeru se spremeni.

Primer 10:
Recimo, da na začetku 24. žarnica gori. Če pritisnemo na vsako drugo stikalo, nato na vsako peto in potem na vsako šesto, bo na koncu 24. žarnica prav tako gorela. Seznam sprememb je zdaj $\{2,5,6\}$, število 24 pa je deljivo z dvema, torej se stanje žarnice ne bo spremenilo.

Ali bi lahko iz končnega stanja in seznama sprememb izračunali začetno stanje? To ni težko premisliti. **Izvedena transformacija se izniči tako, da jo ponovimo.** Torej h končnemu stanju prištejemo vse transformacije, na koncu pa zapišemo ostanek pri deljenju z 2.

Primer 11:
Če smo do končnega stanja prišli s transformacijama T_5, T_7, potem je začetno stanje enako

$$S_0 = (S_k + T_5 + T_7)(mod2)$$

Tu je konkreten primer računanja

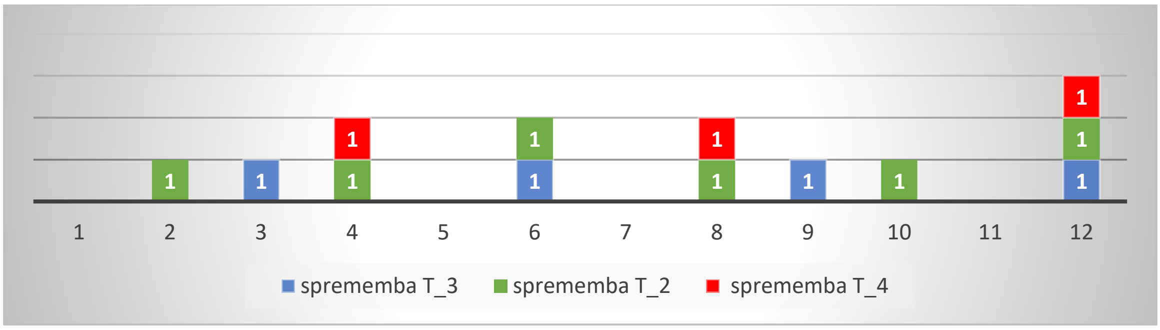
$$\begin{array}{r} S_K = (0,1,0,0,1,1,1,0,1,1,1,1,1,1,1) \\ +\,T_7 = (0,0,0,0,0,0,1,0,0,0,0,0,0,1,0) \\ +\,T_5 = (0,0,0,0,1,0,0,0,0,1,0,0,0,0,1) \\ \hline S_0 = (0,1,0,0,0,1,0,0,1,0,1,1,1,0,0) \end{array}$$

2.6. ZAPOREDJA ŽARNIC

Recimo, da so na začetku vse žarnice ugasnjene in poznamo seznam vseh sprememb, ki jih bomo izvedli. Zanima nas, kakšne lastnosti ima končno stanje. Hitro ugotovimo, da je zaporedje enk in ničel ponavljajoče. Vsaka sprememba T_n učinkuje na vse večkratnike števila n, vse transformacije iz seznama pa skupaj učinkujejo tako, da se ustvari eno **podzaporedje**, ki se potem ponavlja v neskončnost. Dolžino tega ponavljajočega dela bomo označili z d in je enaka najmanjšemu skupnemu večkratniku vseh števil iz seznama sprememb.

Seznam sprememb: $\{a, b, c, \ldots\}$
Dolžina ponavljajočega dela: $d = v(a, b, c, \ldots)$

Primer 12:
Naj bodo vse žarnice v začetku ugasnjene, nato pa zaporedoma izvedemo T_2, T_3, T_4. Te učinkujejo na večkratnike števila 2, 3 in 4, prvič pa vse tri delujejo na žarnico, ki je označena z zaporednim številom, enakim najmanjšemu večkratniku teh števil. Ponavljajoči del ima zato dolžino 12.
Na naslednji sliki je vsaka sprememba pri posamezni žarnici označena z ena, skupni učinek (sprememba) pa nastopi le, če je pri neki žarnici liho število enk.



Tu sta še začetno in končno stanje žarnic iz tega primera.

Vidimo, da je na pogled zelo težko opaziti dele v zaporedju, ki se ponavljajo. Zlasti, če so daljši. Periodičnost veliko lažje razumemo, kot pa opazimo.

Ali pride do ponavljajočih kosov tudi, če je začetno stanje žarnic poljubno? Ne, to lahko potrdimo s konkretnim primerom ali pa premislimo splošno. Če bi bilo končno stanje žarnic periodično, nato pa bi še enkrat izvedli vse transformacije, ki so prav tako periodične, bi tudi rezultat bil periodičen. Pa ni, saj je enak začetnemu stanju žarnic. Poglejmo še primer.

Primer 13:
Začetno stanje je neperiodično, končno prav tako.

$$\begin{aligned} S_0 &= (1,0,1,0,0,0,0,0,0,0,0,0,0,0,0 \dots) \\ +\,T_2 &= (0,1,0,1,0,1,0,1,0,1,0,1,0,1,0, \dots) \\ +\,T_3 &= (0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0 \dots) \\ \hline S_K &= (1,1,0,1,0,0,0,1,1,1,0,0,0,1,1, \dots) \end{aligned}$$

Zdaj lahko pogledamo kake posebne primere končnih stanj žarnic. Recimo, da so na začetku vse žarnice ugasnjene.

Primer 14:
Seznam sprememb naj bo $\{2,4\}$.
Končno stanje je potem: 0100 0100 0100 0100 0100 …

Primer 15:
Seznam sprememb naj bo $\{2,4,8\}$.
Končno stanje je potem: 01000101 01000101 01000101 …

Primer 16:
Seznam sprememb naj bo $\{2,3,4,5,6 \dots\}$ (neskončno mnogo sprememb).
Katere žarnice so na koncu prižgane? Zapišimo vse možnosti glede na vrstni red posamezne žarnice. Stanje se zagotovo spremeni tam, kjer so:

- praštevila (ta so deljiva le z enim številom iz seznama sprememb in se žarnici zato stanje spremeni),
- produkt dveh različnih praštevil (ker je deljiv s tremi števili iz seznama),
- produkt treh različnih praštevil (ker je deljiv s sedmimi števili iz seznama).

Tako je najbrž za vse produkte različnih praštevil. Ugasnjene ostanejo žarnice pri kvadratih praštevil, saj je p^2 deljiv le s p in p^2 iz seznama.

Ugasnjene ostanejo tudi žarnice pri kvadratih naravnih števil iz podobnega razloga.

Kako bi to pojasnili splošno?
Vsako naravno število lahko zapišemo kot produkt praštevil oziroma njihovih potenc (razcep na praštevila).

$$n = p_1^k \cdot p_2^l \cdot \dots p_r^s$$

Število njegovih deliteljev bi potem zmanjšali za 1 in pogledali, če je to število sodo ali liho. Splošno rešitev smo našli v teoriji števil – o številu deliteljev danega števila. Zgoraj zapisano število n ima $(k+1) \cdot (l+1) \cdot \dots \cdot (s+1)$ deliteljev. Pri kvadratih naravnih števil so stopnje v razcepu soda števila, število je deliteljev potem liho mnogo. Če to število zmanjšamo za 1 (ker enke ni na seznamu), dobimo sodo število, torej se stanje žarnice ne bo spremenilo. Poglejmo na primeru.

Primer 17:
Koliko deliteljev ima število $3^6 \cdot 7^2$?
Delitelje dobimo tako, da izberemo poljubno število trojk (od 0 do 6) in poljubno število sedmic (od 0 do 2). Torej imamo skupaj možnosti $7 \cdot 3 = 21$.
Deliteljev je torej 21.

Pojavi se še eno zanimivo vprašanje. Ali lahko iz končanega zaporedja sklepamo o izvedenih transformacijah, torej o seznamu sprememb? Nekaj možnosti imamo, če je zaporedje žarnic periodično. Premislimo ob primeru.

Primer 18:
Naj bo znano končno stanje žarnic, za katerega ugotovimo, da je dolžina periode 18. To je skupni večkratnik nekaj različnih števil, ki tvorijo seznam sprememb.

$$S_k = (0,1,1,1,0,0,0,1,0,1,0,0,0,1,1,1,0,1,0,1,1,1,0,0,0,1,0,1,0,0,0,1\ \dots)$$

Lažje bomo opazovali, če k temu zapišemo še vrstni red žarnic.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	1	1	0	0	0	1	0	1	0	0	0	1	1	1

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
0	1	0	1	1	1	0	0	0	1	0	1	0	0	0	1

Če je prva žarnica ugasnjena, potem med spremembami ni T_1. Nato pogledamo, katera manjša števila imajo enice. Iz tega izvemo, da sta v našem primeru zagotovo bili izvedeni transformaciji T_2in T_3. Zatem pa pogledamo žarnice pri njunih večkratnikih do števila 18. Če bi bila pri 6 žarnica prižgana, bi med spremembami bila tudi T_6.
Kaj pa, če bi se zaporedje začelo z 1? Tedaj bi med spremembami zagotovo bila tudi T_1, ki spremeni stanje vseh žarnic. V tem primeru bi opazovali ugasnjene žarnice.

Ali to pomeni, da lahko vedno določimo seznam sprememb iz končnega periodičnega stanja? Premislimo splošno. Tu predpostavimo, da je zaporedje dobljeno le s spremembami T_n. V poglavju 2.5 smo ugotovili, da lahko iz končnega stanja s ponovno izvedbo sprememb pridemo v začetno stanje. To lahko naredimo tudi postopoma, s preizkušanjem smo prišli do zanimivega postopka.

Algoritem za pridobitev seznama sprememb:

- Najprej pogledamo, če gori prva žarnica. Če gori, stanju prištejemo T_1 in jo izničimo, hkrati damo število 1 na seznam sprememb.
- Potem poiščemo naslednje število n, kjer žarnica gori, in stanju prištejemo T_n , število n damo na seznam sprememb.
- To ponavljamo, dokler se vse ne izniči. Tako dobimo seznam sprememb.

Zapišimo to kot končno ugotovitev, v nadaljevanju pa je še konkreten primer.

Periodično končno stanje žarnic enolično določa seznam sprememb, če je dobljeno le s spremembami T_n.

Primer 19:
Pri končnem stanju po vrsti od leve pogledamo, katero stanje je 1. V našem primeru je to pri številu 2. S tem določimo, da je prva sprememba T_2. Nato prištejemo to spremembo končnemu stanju in s tem izničimo spremembo T_2 in začnemo iskati novo stanje z vrednostjo 1. Pri spodnjem zapisu je to stanje 3. Transformacijo T_3 dodamo med naše spremembe in jo prištejemo k že enkrat spremenjenemu stanju in tako naprej, dokler se vse ne izniči. Seznam sprememb je potem $\{2,3,9\}$ kot kažejo računi.

$$\begin{aligned} S_K &= (0,1,1,1,0,0,0,1,0,1,0,0,0,1,1,1,0,1,0,1,1,1,0,0,0,1,0,1) \\ S_K + T_2 &= (0,0,1,0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0) \\ S_K + T_2 + T_3 &= (0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0) \\ S_K + T_2 + T_3 + T_9 &= (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) \end{aligned}$$

Pojavi se še eno zanimivo vprašanje. Katera so **možna zaporedja** glede na dolžino periodičnega dela? V spodnji tabeli smo zapisali vse možnosti za nekaj začetnih dolžin period (d).

Možna periodična zaporedja:

d	seznam sprememb	periodični del	zaporedje
1	$\{1\}$ $\{0\}$	1 0	11111111 … 0000000 …
2	$\{2\}$ $\{1,2\}$	01 10	01 01 01 … 10 10 10 …
3	$\{3\}$ $\{1,3\}$	001 110	001 001 001 … 110 110 110 …
4	$\{4\}$ $\{1,4\}$ $\{2,4\}$ $\{1,2,4\}$	0001 1110 0100 1011	0001 0001 … 1110 1110 … 0100 0100 … 1011 1011 …
5	$\{5\}$ $\{1,5\}$	00001 11110	00001 00001 … 11110 11110 …
6	$\{6\}$ $\{1,6\}$ $\{2,6\}$ $\{3,6\}$ $\{2,3\}$ $\{1,2,3\}$ $\{1,2,6\}$ $\{1,3,6\}$ $\{2,3,6\}$ $\{1,2,3,6\}$	000001 111110 010100 001000 011100 100011 101011 110111 011101 100010	000001 000001 … 111110 111110 … 010100 010100 … 001000 001000 … 011100 011100 … 100011 100011 … 101011 101011 … 110111 110111 … 011101 011101 … 100010 100010 …

Opazimo, da v tabeli ni vseh periodičnih zaporedji.
Pri dolžini periode 3 izpadeta dve zaporedji 100 100 100 … in 011 011 011 …
Pri dolžini periode 4 pa izpade kar 12 zaporedij, na primer 1000 1000 1000 … in 1101 1101 1101 …

Zakaj zadnje zaporedje ni možno? V tem primeru je bila zagotovo izvedena T_1, prav tako T_3, T_6, T_7, … kot prikazujejo naslednji zapisi.

Dolžina: 4
Periodični del: 1101

$$\begin{aligned} & 1101\ 1101 \ldots \rightarrow \{1\} \\ +T_1 &= 0010\ 0010 \ldots \rightarrow \{1,3\} \\ +T_3 &= 0000\ 0110 \ldots \rightarrow \{1,3,6\} \\ +T_6 &= 0000\ 0010 \ldots \rightarrow \{1,3,6,7\} \\ +T_7 &= 0000\ 0000 \ldots \end{aligned}$$

Najmanjši skupni večkratnik seznama sprememb je 42, kar je v nasprotju z dolžino periode, ki je 4. Torej zgornje zaporedje pri naših spremembah ni možno.

Vsa periodična zaporedja enk in ničel niso pridobljena s spremembami T_n.

Zaradi tega smo popravili prej zapisan algoritem, ki bo določil, ali je nek periodičen del možen v dani dolžini periode in bo v tem primeru izpisal tudi seznam sprememb.

Izboljšan algoritem za pridobitev seznama sprememb:

- Vnos dolžine periode d
- Vnos periodičnega dela zaporedja
- Zapis zaporedja v dolžini 2 d
- Ponavljaj naslednje korake, dokler se vse ne izniči:
  - Poiščemo najmanjše mesto n, kjer žarnica gori
  - Dodamo n na seznam sprememb
  - Stanju prištejemo T_n, da se sprememba izniči
- Izračunaj najmanjši skupni večkratnik števil iz seznama sprememb (v)
- Primerjaj v in d
- Če se v in d ujemata, je opazovana perioda možna in se izpiše seznam sprememb, sicer pa ne.

Odprto ostane vprašanje, če lahko do seznama pridemo tudi v primeru, ko končno stanje ni periodično ali pa če je začetno stanje različno od (0,0,0,…). Prav tako bi bilo zanimivo napisati programček, ki bi prebral zaporedje in vrnil dolžino periodičnega dela.

3. EMPIRIČNI DEL

V programu Python smo izdelali nekaj programov, s katerimi smo preizkusili ugotovitve iz teoretičnega dela. Pognali smo jih na spletni strani repl.it. V nadaljevanju so zaslonski posnetki.

Prvi program ustvari naključno zaporedje enk in ničel. Ima tri vhode; število sprememb, seznam sprememb (posamičen vnos) in zaporedno število izbrane žarnice. Na koncu nam izpiše končno stanje izbrane žarnice.

main.py

```python
from random import randint

#nardi random list
zarnice = []
for i in range(200):
    zarnice.append(randint(0, 1))

print(zarnice)

#kopira random list u koncn list
zarnice_konc = zarnice

katera = int(input("Katera žarnica te zanima? "))
stevilo_sprememb = int(input("Število sprememb: "))

#prazn list za spremembe
list = []
for i in range(200):
    list.append(0)

print("Začetna vrednost na " + str(katera) + "." + " mestu je: " + str(zarnice_konc[katera - 1]))

print("Vpisuj spremembe po vrsti.")

for n in range(stevilo_sprememb):
    sprememba = int(input("Sprememba: "))
    for i in range(200):
        if (i + 1) % sprememba == 0:
            list[i] = list[i] + 1

for j in range(200):
    zarnice_konc[j] = (zarnice_konc[j] + list[j]) % 2

print('Končna vrednost na ' + str(katera) + '.' + ' mestu je: ' + str(zarnice_konc[katera - 1]))
```

```
>_ Console
[0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, [illegible]
 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1,
 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1,
 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1,
 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1,
 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0,
 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1,
 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
Katera žarnica te zanima? 153
Število sprememb: 3
Začetna vrednost na 153. mestu je: 1
Vpisuj spremembe po vrsti.
Sprememba: 2
Sprememba: 3
Sprememba: 9
Končna vrednost na 153. mestu je: 1
```

Drugi program ne spreminja celega zaporedja, temveč le pogleda začetno stanje n-te žarnice in ga spremeni glede na število deliteljev iz seznama sprememb.

main.py

```python
n = int(input('vpišite začetno stanje n-te žarnice(0 ali 1): '))
katera = int(input("Katera žarnica te zanima?: "))
stevilo_sprememb = int(input("Število sprememb: "))

deljitelji = 0
for i in range(stevilo_sprememb):
    sprememba = int(input("Sprememba: "))
    if katera % sprememba == 0:
        deljitelji = deljitelji + 1

if deljitelji % 2 == 1:
    n = (n + 1) % 2

print('Končna vrednost na ' + str(katera) + '. mestu je:' + str(n))
```

```
>_ Console
vpišite začetno stanje n-te žarnice(0 ali 1): 1
Katera žarnica te zanima?: 153
Število sprememb: 3
Sprememba: 2
Sprememba: 3
Sprememba: 9
Končna vrednost na 153. mestu je:1
```

Tretji program izvede izboljšan algoritem za ugotavljanje seznama sprememb iz danega periodnega dela. Vmes preveri, če je periodični del sploh mogoč glede na dolžino periode.

main.py

```
import math
def findlcm(a):
    lcm=int(a[0])
    for i in range(1,len(a)):
      lcm=lcm*(int(a[i]))//math.gcd(lcm,int(a[i]))
    return lcm

d = int(input('vpišite dolžino periode: '))
perioda = str(input('Vpišite periodni del zaporedja: '))
if not len(perioda) == d:
    print('PERIODA NI PRAVE DOLŽINE!')
else:
    zaporedje = []

    for t in range(2):
        for o in range(len(perioda)):
            zaporedje.append(int(perioda[o]))

    trans = []
    list = []

    for k in range(len(zaporedje)):
        if int(zaporedje[k]) == 1:
            trans.append(k + 1)
            for i in range(len(zaporedje)):
                if (i + 1) % (k + 1) == 0:
                    list.append(1)
                else:
                    list.append(0)

            for p in range(len(zaporedje)):
                zaporedje[p] = (list[p] + zaporedje[p]) % 2
        for h in range(len(zaporedje)):
            list = []


    if findlcm(trans) == d:
        print('Seznam sprememb: ' + str(trans))
    else:
        print('PERIODA NI MOGOČA!')
```

Ta program smo pognali na dveh primerih (pri enem je periodični del mogoč, pri drugem pa ne).

```
>_ Console    Shell
vpišite dolžino periode: 6
Vpišite periodni del zaporedja: 010100
Seznam sprememb: [2, 6]
```

```
>_ Console    Shell
vpišite dolžino periode: 4
Vpišite periodni del zaporedja: 1101
PERIODA NI MOGOČA!
```

4. ZAKLJUČEK

Zaporedja so pomembna matematična orodja, ki se pojavijo na mnogih področjih matematike in tudi v našem vsakdanjem življenju. V raziskovalni nalogi smo se osredotočili na preučevanje zaporedji v povezavi z žarnicami. Da bi lažje razumeli ta zaporedja, smo na začetku morali priti do nekaterih osnovnih spoznanj. Ugotovili smo, da vrstni red sprememb pri žarnicah ni pomemben. Sodo število enakih sprememb ne povzroči spremembe, če pa imamo liho število enakih sprememb, je enako kot bi spremembo naredili samo enkrat. V nadaljevanju smo se naučili računati končno stanje s seštevanjem po modulu 2, tako da za vsako žarnico na poljubnem mestu vemo v kakšnem stanju je. Ob koncu naloge smo iz končnega stanja žarnic uspeli dobiti seznam sprememb, če nam je bil na začetku neznan. Našli smo tudi algoritem, ki presodi, če je neko periodično zaporedje dobljeno z opazovanimi spremembami.

Razmišljanja in ugotovitve smo ilustrirali s konkretnimi primeri. Izdelali smo tudi nekaj programov v Pythonu, s katerimi smo preizkusili naše ugotovitve.

Kaj smo se naučili?

- novi zapisi
- računanje po komponentah in modulu 2
- program Python

Kaj smo ugotovili?

- Iz začetnega stanja in seznama sprememb lahko ugotovimo končno stanje.
- Iz končnega stanja lahko izračunamo seznam sprememb, če poznamo tudi začetno stanje in je dobljeno le s spremembami T_n
- Vsa periodična stanja nisi dobljena s temi spremembami.

Kaj ostaja odprto?

- Vse, kar je povezano z neperiodičnim stanjem žarnic.
- Vse v zvezi z neskončnimi seznami sprememb.

5. VIRI

- Enciklopedija števil, Jože Grasselli, strani 453 in 482 , 1. izdaja, 1. natis ,Ljubljana : Založba DMFA, 2008.
- Učbenik Planum Novum : matematika za gimnazije, Gregor Pavlič in drugi, 1. izdaja, 2. natis, strani 94-133, Ljubljana : Modrijan, 2013.
- Učbenik Linea nova : matematika za gimnazije, Gregor Pavlič in drugi, 1. izdaja, 4. natis, strani 34-53, Ljubljana : Modrijan, 2012.
- Wikipedia: https://sl.wikipedia.org/wiki/Delitelj (Prevzeto: 1.3.2023)
- YouTube: https://www.youtube.com/@programmingwithmosh (Prevzeto: 1.3.2023)