UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Nevena Pantelić

Jangovi dijagrami slučajnih permutacija

master rad

Beograd, 2023.

Mentor:

dr Miodrag Živković, redovni profesor u penziji
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

dr Filip Marić, redovni profesor
Univerzitet u Beogradu, Matematički fakultet

dr Vesna Marinković, docent
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane:

Naslov master rada: Jangovi dijagrami slučajnih permutacija

Rezime: Svaka od $n!$ permutacija skupa veličine n može se podeliti u cikluse, čime je određena jedna particija broja n. Particija se često predstavlja Jangovim dijagramom.

Osnovni cilj je odrediti prosečne vrednosti statistika dijagrama za svih $n!$ permutacija reda n, što se može uraditi na nekoliko načina:

- tačno, za male vrednosti broja n (računajući prosek po svim permutacijama)
- tačno, za nešto veće n (računajući prosek po svim particijama)
- približno (generišući slučajne permutacije reda n i računajući proseke statistika za generisane permutacije)

U ovom radu su analizirane statistike (širina, visina, punoća i asimetrija) Jangovih dijagrama koji odgovaraju razlaganjima permutacija reda n u cikluse.

Ključne reči: permutacija, particija, Jangov dijagram, statistika, simetrična grupa, metod Monte Karlo

Sadržaj

1 Uvod

Neka je n prirodan broj. Permutacije skupa od n elemenata formiraju simetričnu grupu S_n. Drugim rečima, S_n predstavlja grupu svih mogućih uređenja n objekata. Ovakva grupa ima $n!$ elemenata. Na primer, grupa S_3 ima $3! = 6$ elemenata, koji predstavljaju sve moguće permutacije skupa od 3 elementa.

Svaki element grupe S_n predstavlja permutaciju, tj. bijektivno preslikavanje skupa $\{0, 1, 2, ..., n - 1\}$ u samog sebe. Jedan od elemenata simetrične grupe je i **indentička transformacija** (identitet), koja sve elemente slika u sebe same.

Na primer, svi elementi grupe S_3 su:

- $\{1, 2, 3\} \longrightarrow \{1, 2, 3\}$
- $\{1, 2, 3\} \longrightarrow \{2, 1, 3\}$
- $\{1, 2, 3\} \longrightarrow \{3, 2, 1\}$
- $\{1, 2, 3\} \longrightarrow \{1, 3, 2\}$
- $\{1, 2, 3\} \longrightarrow \{2, 3, 1\}$
- $\{1, 2, 3\} \longrightarrow \{3, 1, 2\}$

Simetrične grupe imaju bitnu ulogu u mnogim granama matematike, kao što su teorija grupa, kombinatorika, algebra. Osim matematičkih disciplina, koriste se i u fizici, kriptografiji i drugim oblastima računarstva. Detaljnije pogledati u [2].

Svaka permutacija može se podeliti u cikluse.

Primer 1: za $n = 8$ permutacija skupa $\{0, 1, 2, 3, 4, 5, 6, 7\}$ prikazana u tabeli 1 sastoji se od **tri** ciklusa:

- $0 \longrightarrow 3 \longrightarrow 2 \longrightarrow 4 \longrightarrow 7 \longrightarrow 0$
- $1 \longrightarrow 6 \longrightarrow 1$
- $5 \longrightarrow 5$

a	0	1	2	3	4	5	6	7
b	3	6	4	2	7	5	1	0

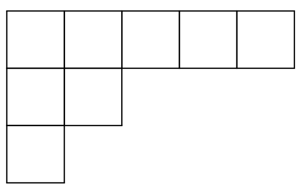
Tabela 1: Permutacija koja cifre a slika u cifre b

Particija prirodnog broja n je neki od načina njegovog predstavljanja u obliku zbira prirodnih brojeva $n = x_1 + x_2 + ... + x_y$, $x_1 \geq x_2 \geq ... \geq x_y$ (detaljnije u [3] i [6]).

Dužine ciklusa određuju jednu od particija broja n. U primeru 1 ciklusi određuju particiju $\mathbf{8 = 5 + 2 + 1}$ broja 8.

Particije prirodnih brojeva se često predstavljaju **Jangovim dijagramima**. Kako je sabiranje komutativna operacija, particije se zapisuju tako da sabirci budu u nerastućem poretku (iste particije imaju isti dijagram). Jangov dijagram je struktura čiji redovi imaju onoliko ćelija kolika je vrednost trenutnog sabirka. Dakle, za particiju $n = x_1 + x_2 + ... + x_y$, $x_1 \geq x_2 \geq ... \geq x_y$, prvi red ima x_1 ćelija, drugi red x_2 ćelija,..., poslednji x_y ćelija. Ukupno ima y redova.

Za particiju $8 = 5 + 2 + 1$ iz primera 1, prvi red Jangovog dijagrama ima 5 ćelija, drugi 2, i poslednji red jednu ćeliju. Redova u Jangovom dijagramu ima koliko i ciklusa u permutaciji, odnosno sabiraka u particiji. Za ovu particiju Jangov dijagram ima tri reda (videti sliku 1).



Slika 1: Jangov dijagram particije $8 = 5 + 2 + 1$

Širina Jangovog dijagrama predstavlja dužinu najdužeg ciklusa, tj. prvi (najveći) sabirak u particiji ($x = x_1$), a **visina** Jangovog dijagrama predstavlja broj ciklusa, tj. broj sabiraka u particiji (y). Za dijagram u slučaju particije iz primera 1 je $x = 5$, $y = 3$.

Jangov dijagram je zapravo jedan deo pravougaonika sa stranicama x i y. Udeo površine koju zauzima Jangov dijagram od ukupne površine pravougaonika naziva se **punoća** dijagrama. Kako Jangov dijagram zauzima n ćelija pravougaonika $x \times y$, to se njegova punoća računa na osnovu izraza: $\lambda = \frac{n}{xy}$. Za dijagram u slučaju particije iz primera 1 $\lambda = \frac{8}{15}$.

Osim ove tri osobine Jangovog dijagrama, razmatra se i četvrta, **asimetrija** i ona se računa na osnovu izraza $\mu = \frac{y}{x}$. Za dijagram u slučaju particije iz primera 1 asimetrija je: $\mu = \frac{3}{5}$.

U ovom radu razmatraju se navedene karakteristike Jangovih dijagrama. Interesantno je videti koji od dijagrama se najčešće javlja među svim permutacijama (koji je najbolji predstavnik te grupe permutacija). Kako skup od n elemenata ima $n!$ permutacija, za veliko n proseke po svim permutacijama nije realno izračunati čak ni uz pomoć računara. Za manje n ($n \leq 10$) mogu se lako odrediti karakteristike crtanjem Jangovih dijagrama za sve permutacije. Međutim, kako broj n raste, tako raste i broj mogućih načina njegovog particionisanja p_n (videti tabelu 2), ali mnogo sporije od $n!$. Zbog toga se tačne vrednosti statisika mogu odrediti za nešto veće n težinskim usrednjavanjem njihovih vrednosti za pojedine particije.

n	1	2	3	4	5	6	7	8	9	10
p_n	1	2	3	5	7	11	15	22	30	42

Tabela 2: Broj particija broja n

Za velike n prosečne vrednosti statistika mogu se proceniti heurističkim eksperimentom. Da bi se dobila asimptotska procena statistika, koristi se sledeći postupak: na početku se generiše slučajna permutacija skupa veličine n, odrede se statistike, zatim se eksperiment ponavlja veliki broj puta i računa se prosek po generisanim permutacijama.

Drugim rečima moguće je koristiti metod Monte Karlo, generisati slučajne permutacije i računati proseke statistika za te permutacije. Metod Monte Karlo se koristi za rešavanje problema koji imaju konkretno numeričko rešenje. Čitav metod sastoji se od narednih nekoliko koraka:

- definisanje problema i domena ulaznih podataka
- generisanje niza slučajnih objekata iz tog domena
- računanje statistika nad tim nizom
- davanje procene rešenja problema na osnovu dobijenih rezultata iz prethodnog koraka

Ovaj metod je izuzetno koristan, posebno za rad sa složenim sistemima i ima široku primenu. Osim računarstva i matematike, koristi se i u fizici, biologiji, dizajnu...(detaljnije u [8])

2 Asimptotske vrednosti statistika

Dok su u glavi 1 predstavljeni osnovni teorijski pojmovi, ova glava ima za cilj izračunavanje tačnih (ukoliko je moguće) i prosečnih statistika, ukoliko je previše komplikovano izračunati tačne. Ova izračunavanja se ekperimentalno proveravaju u narednoj glavi.

2.1 Statistike Jangovih dijagrama za male vrednosti n

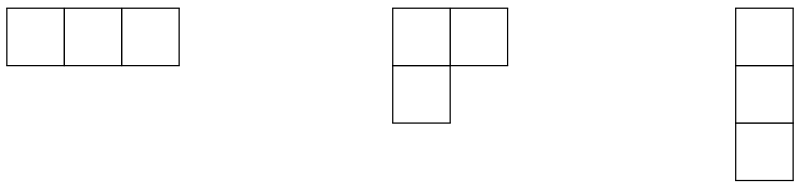
Kako za malo n nije teško nacrtati Jangov dijagram, na početku ovog dela je to i urađeno za nekoliko prirodnih brojeva (analogno bi se moglo uraditi i za neke veće brojeve).

Za početak, potrebno je uvesti još neke oznake. Sa D se predstavlja Jangov dijagram. Kako? Piše se u obliku proizvoda, gde su činioci zapravo sabirci iz particije, a stepeni su brojevi njihovih pojavljivanja u particiji:

$$D = x_1^{z_1} \cdot x_2^{z_2} \cdots x_y^{z_y}$$

Sabirak x_1 se pojavljuje z_1 puta, sabirak x_2 se pojavljuje z_2 puta...sabirak x_y se pojavljuje z_y puta.

Primer 2: za $n = 3$ postoje tri particije ($3 = 3$; $3 = 2 + 1$; $3 = 1 + 1 + 1$). U slučaju particije $3 = 3$ sabirak 3 se pojavljuje jednom, pa je $D = 3^1 = 3$. U drugom slučaju ($3 = 2 + 1$) sabirak 2 se pojavljuje jednom i sabirak 1 se pojavljuje jednom, pa je $D = 2^1 \cdot 1^1 = 2 \cdot 1$. U poslednjem slučaju $(3 = 1 + 1 + 1)$, sabirak 1 se pojavljuje tri puta, pa je $D = 1^3$. Kako postoje tri particije, postoje i tri dijagrama, koji su prikazani na slici 2.1



$D = 3^1 = 3$ $\quad$ $D = 2 \cdot 1$ $\quad$ $D = 1^3$

Slika 2.1: Jangovi dijagrami za sve particije broja 3

Posmatranjem Jangovih dijagrama, može se uočiti i sledeće: u prvom slučaju postoje 3 ćelije u jednom redu, pa je $D = 3^1 = 3$. U drugom slučaju postoje 2 ćelije u jednom redu i jedna ćelija u još jednom redu, pa je $D = 2^1 \cdot 1^1 = 2 \cdot 1$. U poslednjem slučaju postoji po jedna ćelija u tri reda, pa je $D = 1^3$.

Osim ove, koristi se i oznaka za broj permutacija sa datim Jangovim dijagramom D i obeležava se sa $N(D)$. Broj $N(D)$ zapravo predstavlja broj permutacija koje odgovaraju istom Jangovom dijagramu (tj. particiji - na početku je bilo reči o tome da se particije posmatraju u nerastućem poretku zbog preklapanja, odakle se i naslućuje potreba za ovim podatkom). Koliko ima permutacija reda n sa particijom $x_1^{z_1} \cdot x_2^{z_2} \cdots x_y^{z_y}$? Ako se ciklusi poređaju redom: postoji z_1 ciklusa duzine x_1, z_2 ciklusa duzine x_2 ... z_y ciklusa duzine x_y, n brojeva može se rasporediti na $n!$ načina, ali ako bilo koji ciklus dužine x_1 bude ciklički ispermutovan, dobija se ista permutacija. Zapravo, svaki ciklus dužine x_1 se tako može ispermutovati na x_1 načina. Iz ovoga sledi da se broj $n!$ mora podeliti sa $x_1^{z_1}$. Osim toga, permutacija se ne menja ni ako se ispermutuje z_1 ciklusa dužine x_1, odakle se može zaključiti da je ukupan broj permutacija $n!$ neophodno podeliti i brojem $z_1!$. Isto važi za x_2,..., x_y. Detaljnije u [4]. Dakle,

$$N(D) = \frac{n!}{x_1^{z_1} \cdot z_1! \cdot x_2^{z_2} \cdot z_2! \cdots x_y^{z_y} \cdot z_y!}$$

U odeljku 2.1 može se videti proces izračunavanja $N(D)$ (za $n = 4$, $n = 5$...).

Klase konjugovanih elemenata (klase elemenata sa istom ciklusnom strukturom) simetrične grupe S_n predstavljaju skupove permutacija koje određuju istu particiju. Broj klasa konjugovanih elemenata grupe S_n je jednak broju particija broja n. Na primer, za $n = 5$, simetrična grupa S_5 ima $5! = 120$ elemenata, što je ukupan broj permutacija skupa od 5 elemenata, a klasa konjugacije ove grupe ima 7, koliko ima particija broja 5.

U nastavku sledi izračunavanje statistika za prvih nekoliko prirodnih brojeva.

- $n = 3$: U ovom slučaju već su bili crtani Jangovi dijagrami (bilo ih je 3). Međutim, permutacija reda 3 ukupno postoji $3! = 6$. Grupa permutacija ova tri elementa može se posmatrati kao grupa simetrija jednakostraničnog trougla, posmatrajući ove permutacije kao permutacije temena trougla. Ova grupa sadrži **dve** rotacije reda tri, **tri** osne simetrije reda dva (refleksija u odnosu na tri visine trogula) i (**jednu**) identičku transformaciju.

  Podebljani brojevi u prethodnom pasusu zapravo predstavljaju vrednosti broja $N(D)$ za različite dijagrame D.

  U tabeli 3 prikazane su vrednosti statistika za $n = 3$:

D	$N(D)$	x	y	λ	μ
3	2	3	1	1	0.33
$2 \cdot 1$	3	2	2	0.75	1
1^2	1	1	3	1	3

Tabela 3: Vrednosti statistika za $n = 3$

  Prosečne vrednosti statistika mogu se izračunati množenjem vrednosti sa brojem pojavljivanja takvog dijagrama ($N(D)$) i deljenjem sa ukupnim brojem permutacija ($n!$):

$$\hat{x} = \frac{3\cdot 2 + 2\cdot 3 + 1}{3!} = \frac{6+6+1}{6} = \frac{13}{6} = 2\frac{1}{6} \approx 2.17$$

$$\hat{y} = \frac{1\cdot 2 + 2\cdot 3 + 3\cdot 1}{3!} = \frac{2+6+3}{6} = \frac{11}{6} = 1\frac{5}{6} \approx 1.83$$

$$\hat{\lambda} = \frac{1\cdot 2 + \frac{3}{4}\cdot 3 + 1\cdot 1}{3!} = \frac{2+\frac{9}{4}+1}{6} = \frac{8+9+4}{4\cdot 6} = \frac{21}{24} = \frac{7}{8} \approx 0.88$$

$$\hat{\mu} = \frac{\frac{1}{3}\cdot 2 + 1\cdot 3 + 3\cdot 1}{3!} = \frac{\frac{2}{3}+3+3}{6} = \frac{2+9+9}{3\cdot 6} = \frac{20}{18} = \frac{10}{9} \approx 1.11$$

- $n = 4$: Grupa permutacija ova četiri elementa može se posmatrati kao grupa simetrija tetraedra.

D	$N(D)$	x	y	λ	μ
4	6	4	1	1	0.25
$3\cdot 1$	8	3	2	0.67	0.67
2^2	3	2	2	1	1
$2\cdot 1^2$	6	2	3	0.67	1.5
1^4	1	1	4	1	4

Tabela 4: Vrednosti statistika za $n = 4$

Kako su dobijene vrednosti za $N(D)$?

U slučaju $D = 4$:

$$N(D) = \frac{4!}{4^1\cdot 1!} = \frac{24}{4} = 6$$

U slučaju $D = 3 \cdot 1$:

$$N(D) = \frac{4!}{3^1 \cdot 1! \cdot 1^1 \cdot 1!} = \frac{24}{3} = 8$$

U slučaju $D = 2^2$:

$$N(D) = \frac{4!}{2^2 \cdot 2!} = \frac{24}{8} = 3$$

U slučaju $D = 2 \cdot 1^2$:

$$N(D) = \frac{4!}{2^1 \cdot 1! \cdot 1^2 \cdot 2!} = \frac{24}{4} = 6$$

U slučaju $D = 1^4$:

$$N(D) = \frac{4!}{1^4 \cdot 4!} = \frac{24}{24} = 1$$

Prosečne vrednosti statistika:

$$\hat{x} = \frac{4 \cdot 6 + 3 \cdot 8 + 2 \cdot 3 + 2 \cdot 6 + 1 \cdot 1}{4!} = \frac{67}{24} = 2\frac{19}{24} \approx 2.79$$

$$\hat{y} = \frac{1 \cdot 6 + 2 \cdot 8 + 2 \cdot 3 + 3 \cdot 8 + 4 \cdot 1}{4!} = \frac{50}{24} = 2\frac{1}{12} \approx 2.08$$

$$\hat{\lambda} = \frac{1 \cdot 6 + \frac{4}{6} \cdot 8 + 1 \cdot 3 + \frac{4}{6} \cdot 6 + 1 \cdot 1}{24} = \cdots = \frac{29}{36} \approx 0.80$$

$$\hat{\mu} = \frac{\frac{1}{4} \cdot 6 + \frac{2}{3} \cdot 8 + 1 \cdot 3 + \frac{3}{2} \cdot 6 + 4 \cdot 1}{4!} = \cdots = \frac{137}{144} \approx 0.95$$

- $n = 5$: Klase konjugovanih elemenata simetrične grupe S_5

D	$N(D)$	x	y	λ	μ
5	24	5	1	1	0.2
$4 \cdot 1$	30	4	2	0.62	0.5
$3 \cdot 2$	20	3	2	0.83	0.67
$3 \cdot 1^2$	20	3	3	0.56	1
$2^2 \cdot 1$	15	2	3	0.83	1.5
$2 \cdot 1^3$	10	2	4	0.62	2
1^5	1	1	5	1	5

Tabela 5: Vrednosti statistika za $n = 5$

Kako su dobijene vrednosti za $N(D)$?

Na primer, za $D = 2 \cdot 1^3$:

$$N(D) = \frac{5!}{2^1 \cdot 1! \cdot 1^3 \cdot 3!} = \frac{120}{12} = 10$$

Prosečne vrednosti statistika:

$\hat{x}$	$\hat{y}$	$\hat{\lambda}$	$\hat{\mu}$
$3\frac{17}{40}$	$2\frac{17}{60}$	$\frac{325}{432}$	$\frac{184}{225}$
≈ 3.42	≈ 2.28	≈ 0.75	≈ 0.82

Tabela 6: Prosečne vrednosti statistika za $n = 5$

- $n = 6$: Klase konjugovanih elemenata simetrične grupe S_6

D	$N(D)$	x	y	λ	μ
6	120	6	1	1	0.17
$5 \cdot 1$	144	5	2	0.6	0.4
$4 \cdot 2$	90	4	2	0.75	0.5
$4 \cdot 1^2$	90	4	3	0.5	0.75
3^2	40	3	2	1	0.67
$3 \cdot 2 \cdot 1$	120	3	3	0.67	1
$3 \cdot 1^3$	40	3	4	0.5	1.33
2^3	15	2	3	1	1.5
$2^2 \cdot 1^2$	45	2	4	0.75	2
$2 \cdot 1^4$	15	2	5	0.6	2.5
1^6	1	1	6	1	6

Tabela 7: Vrednosti statistika za $n = 6$

Prosečne vrednosti statistika:

$\hat{x}$	$\hat{y}$	$\hat{\lambda}$	$\hat{\mu}$
$4\frac{31}{720}$	$2\frac{408}{726}$	$\frac{10353}{14400}$	$\frac{32836}{43200}$
≈ 4.04	≈ 2.57	≈ 0.72	≈ 0.76

Tabela 8: Prosečne vrednosti statistika za $n = 6$

- $n = 7$: Klase konjugovanih elemenata simetrične grupe S_7

D	$N(D)$	x	y	λ	μ
7	720	7	1	1	0.14
$6 \cdot 1$	840	6	2	0.58	0.33
$5 \cdot 2$	504	5	2	0.7	0.4
$5 \cdot 1^2$	504	5	3	0.47	0.6
$4 \cdot 3$	420	4	2	0.87	0.5
$4 \cdot 2 \cdot 1$	630	4	3	0.58	0.75
$4 \cdot 1^3$	210	4	4	0.44	1
$3^2 \cdot 1$	280	3	3	0.78	1
$3 \cdot 2^2$	210	3	3	0.78	1
$3 \cdot 2 \cdot 1^2$	420	3	4	0.58	1.33
$3 \cdot 1^4$	70	3	5	0.47	1.67
$2^3 \cdot 1$	105	2	4	0.87	2
$2^2 \cdot 1^3$	105	2	5	0.7	2.5
$2 \cdot 1^5$	21	2	6	0.58	3
1^7	1	1	7	1	7

Tabela 9: Vrednosti statistika za $n = 7$

Prosečne vrednosti statistika prikazane su u tabeli:

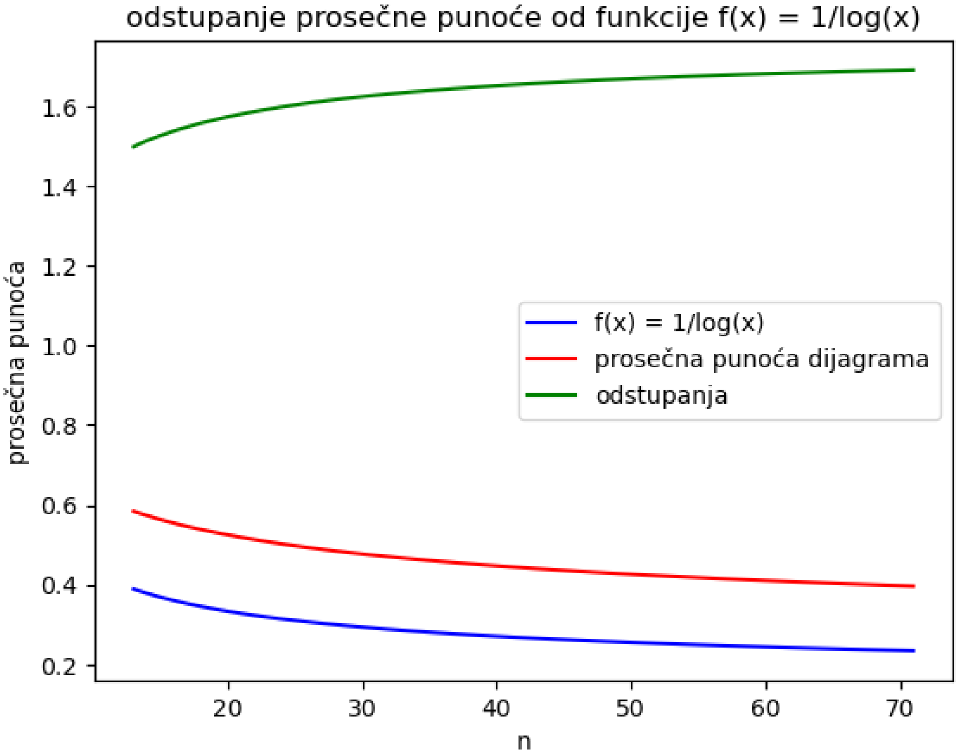
$\hat{x}$	$\hat{y}$	$\hat{\lambda}$	$\hat{\mu}$
≈ 4.68	≈ 2.71	≈ 0.69	≈ 0.70

Tabela 10: Prosečne vrednosti statistika za $n = 7$

U prethodnim tabelama date su prosečne vrednosti statistika (širina, visina, punoća i asimetrija) do $n = 7$, dok je za $n < 100$ efikasnije računati prosečne vrednosti po svim particijama (pogledati odeljak 3.2). Do sada su izračunate statistike tek za nekoliko pvih prirodnih brojeva, međutim već za ove vrednosti se mogu naslutiti neki trendovi.

Neki od zaključaka koje je Arnold izveo u svojoj knjizi [1]:

- prosečna širina dijagrama raste linearno u odnosu na njegovu površinu n: $\hat{x} \approx c_1 n$ (videti slike iz odeljka 3.1 i 3.2)
- prosečna visina dijagrama raste prilično sporo u odnosu na rast njegove površine, otprilike $\hat{y} \approx c_2 \ln n$ (Visina se približno duplira kada se površina kvadrira - videti slike iz odeljka 3.1 i 3.2)
- prosečna punoća dijagrama polako opada kako površina n raste: $\hat{\lambda} \approx \frac{c_3}{\ln n}$ (Punoća se prepolovi kada se površina kvadrira - videti sliku 2.1)
- prosečna asimetrija opada kako n raste, dijagrami postaju „ravniji” - sve niži i širi
  Međutim, geometrijska sredina logaritma asimetrije je $\sigma \approx 0.30$ za $n = 2$ i raste do $\sigma \approx 0.42$ za $n = 7$. Rast veličine σ pokazuje da dijagrami ostaju poprilično asimetrični, čak i za veliko n. Zapravo oni mogu imati širok spektar oblika, uključujući siroke dijagrame za $\mu < 1$ i visoke dijagrame za $\mu > 1$



Slika 2.1 Procena prosečne vrednosti punoće

2.2 Statistike Jangovih dijagrama za velike vrednosti n

Dalji račun bi se mogao nastaviti pomoću računara i za nešto veće n, ali za dosta velike brojeve n, prosek svih $n!$ permutacija nije realno izračunati, na primer za $n = 100$.

U ovom slučaju primenjuje se drugi pristup, koji se sastoji od konstrukcije slučajne permutacije reda n, formiranja particije, koja određuje Jangov dijagram i računanja njegovih karakteristika. Ovaj postupak ponavlja se veći broj puta, kako bi rezultati bili upoređivani i kako bi mogla da se odredi priroda ponašanja ovakvih dijagrama.
Ono što je potrebno obezbediti je uniformna raspodela verovatnoća slučajnih permutacija, a to se dobija na sledeći način:

- Konstruiše se niz $\{0, 1, ..., n - 1\}$
- Počinje se od poslednjeg elementa (sa indeksom $n - 1$), on se menja sa slučajno izabranim elementom niza između indeksa 0 i $n - 1$
- Ponavlja se prethodni korak smanjujući indeks (od $n - 2$ do 1) dok se ne stigne do prvog elementa (sa indeksom 0)

Opisani algoritam naziva se Knutovo (Fišer - Jetsovo) mešanje (videti [5]). Ostaje da se pokaže da ovakav algoritam daje uniformnu raspodelu verovatnoća slučajnih permutacija. Za to je dovoljno pokazati da:

- Svaka permutacija je ostvariva: Postoji niz zamena koji rezultira tom permutacijom
- Svaka permutacija je jednako verovatna: Svaka permutacija ima jednake šanse da bude izabrana

Ova dva uslova je moguće ostvariti korišćenjem uniformne raspodele za izbor indeksa za zamenu, a u programskim jezicima to omogućava generator slučajnih brojeva.

U ovoj glavi nije prikazana konstrukcija slučajnih permutacija (U Arnoldovoj knjizi [1] postoji detaljan opis), već samo izneti njegovi zaključci. Izvršivši računanje statistika za slučajne permutacije, dobio je rezultate prikazane u tabeli 11:

n	D	x	y	λ	μ
16	$7 \cdot 3 \cdot 2^2 \cdot 1^2$	7	6	0.38	0.86
25	$9 \cdot 7 \cdot 5 \cdot 3 \cdot 1$	9	5	0.56	0.56
64	$35 \cdot 15 \cdot 7 \cdot 3 \cdot 2 \cdot 1^2$	35	7	0.26	0.20
100	$42 \cdot 36 \cdot 18 \cdot 2 \cdot 1^2$	42	6	0.40	0.14
100	$90 \cdot 4 \cdot 3 \cdot 2 \cdot 1$	90	5	0.22	0.06
100	$88 \cdot 9 \cdot 1^3$	88	5	0.23	0.06
169	$147 \cdot 13 \cdot 8 \cdot 1$	147	4	0.29	0.03

Tabela 11: Statistike slučajnih permutacija

U prvoj koloni ove tabele nalaze se vrednosti broja n. U drugoj koloni nalazi se Jangov dijagram slučajne permutacije, dok poslednje četiri kolone predstavljaju vrednosti statistika odgovarajućeg Jangovog dijagrama D.

3 Eksperimentalna provera asimptotike

Ova glava posvećena je izračunavanju statistika Jangovih dijagrama u zavisnosti od veličine broja n, koristeći Python kod. Funkcije za računanje statistika su napravljene tako da za ulaz imaju broj n, osim poslednje (za rad sa slučajnim permutacijama, za veoma velike n), koja za ulaz uzima i broj k - broj slučajnih permutacija za generisanje. Za izlaz imaju izračunate statistike Jangovih dijagrama. Za svaku od funkcija opisano je šta radi i po kom algoritmu. Eksperimentima je procenjena zavisnost vremena izračunavanja i relativne greške od n (u slučaju generisanja slučajnih permutacija za $n < 100$).

3.1 Tačne statistike za $n \leq 10$

Za ovako male vrednosti n korišćena je funkcija koja računa proseke statistika Jangovih dijagrama po svim permutacijama. Kako permutacija skupa od 10 elemenata ima 10!, računar može ovaj posao da obavi za prihvatljivo vreme.

Funkcija `statistics_small_n` kao ulazni argument prima prirodni broj n, a onda u listu `partics` upisuje sve moguće particije broja n (za svaku permutaciju skupa od n elemenata se formira particija). Zatim za svaku od particija formira Jangov dijaram i ažurira sume dobijenim statistikama. Na kraju, sume deli sa $n!$, jer toliko ukupno ima permutacija reda n.

```
def statistics_small_n(n):
    x_sum = 0
    y_sum = 0
    lam_sum = 0
    asim_sum = 0
    partics = particije(n)
    for particija in partics:
        (x, y, lam, asim, De, Ne) = Jangov_dijagram(particija, n)
        x_sum += x
        y_sum += y
        lam_sum += lam
        asim_sum += asim
    x_avg = x_sum/faktorijel(n)
    y_avg = y_sum/faktorijel(n)
    lam_avg = lam_sum/faktorijel(n)
    asim_avg = asim_sum/faktorijel(n)
    return x_avg, y_avg,lam_avg, asim_avg
```

Funkcija `statistics_small_n` poziva funkciju `Jangov_dijagram` koja za datu particiju pravi Jangov dijagram. Kako je ulazni argument particija, broj n možemo dobiti sumiranjem svih elemenata particije. Kao što je u uvodu rečeno, x je najveći (prvi) element particije, y je broj particija, dok se punoća, asimetrija, D i $N(D)$ računaju na osnovu izraza opisanih u glavi 1.

```
def Jangov_dijagram(particija, n):
    x = particija[0]
    y = len(particija)
    lam = n/(x*y)
    asim = y/x
    De = D(particija)
    Ne = N(particija)
    return(x, y, lam, asim, De, Ne)
```

Funkcija `statistics_small_n` poziva i funkciju `particije` (videti dodatak A.1) koja za dati broj n vraća particije tog broja koje odgovaraju svim permutacijama skupa od n elemenata (videti dodatak A.2). Za ulaz n ova funkcija prvo formira listu svih permutacija, a zatim za svaku od njih poziva funkciju koja od permutacije formira particiju, sortira njene elemente u nerastućem poretku i dodaje na listu particija koju vraća.

Kako radi ovaj deo programa? Funkcija `print_statistics_small_n` koristi se za ispis vrednosti, čiji potpis se može videti u dodatku A.3.

U slučaju $n = 3$ izlaz je sledeći:

```
print_statistics_small_n(3)
```

```
Prosečne vrednosti su:
širina: 2.17, dužina:  1.83, punoća: 0.88, asimetrija: 1.11
```

Slika 3.11: `print_statistics_small_n`

U sledećoj tabeli prikazane su prosečne vrednosti statistika za $n < 10$, kao i vremena izvršavanja u zavisnosti od broja n. Ova tabela sadrži nešto veće vrednosti n nego tabele iz odeljka 2.1 i može se primetiti da se vrednosti u ovoj tabeli neznatno razlikuju od onih iz 2.1.

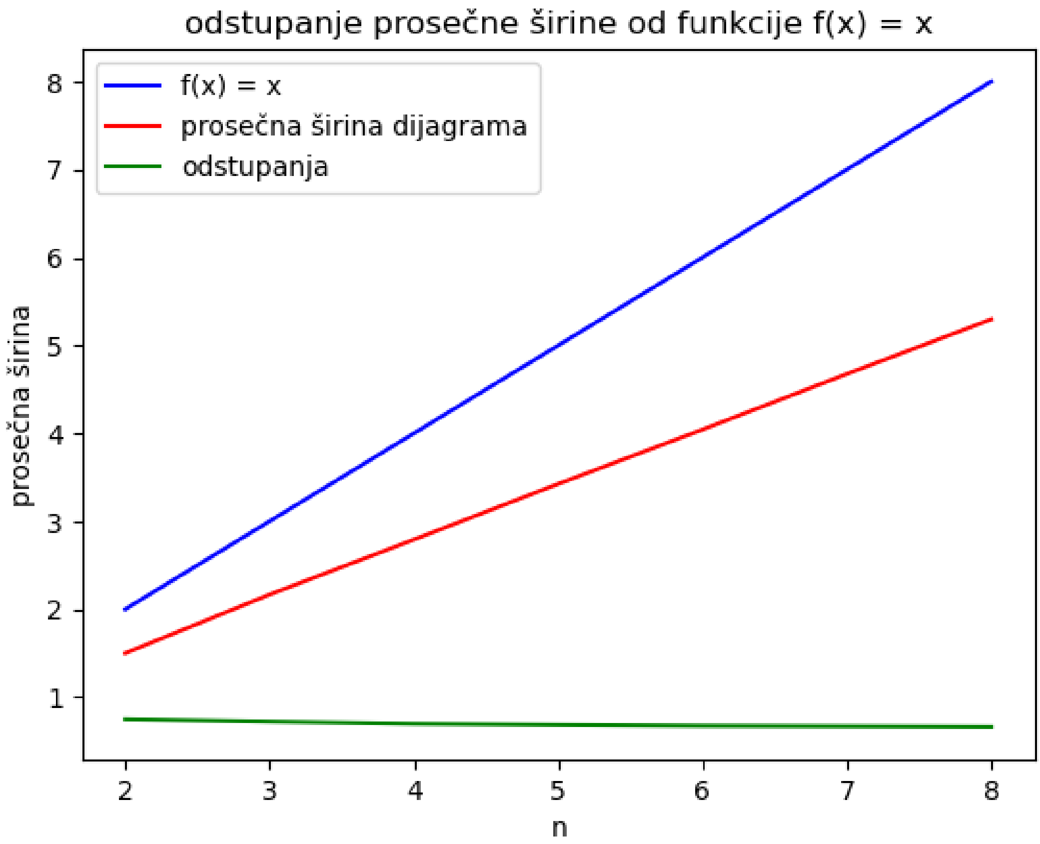
n	$\hat{x}$	$\hat{y}$	$\hat{\lambda}$	$\hat{\mu}$	vreme izvršavanja
4	2.79	2.08	0.81	0.95	$0.001s$
5	3.42	2.28	0.75	0.84	$0.007s$
6	4.04	2.45	0.72	0.76	$0.046s$
7	4.68	2.59	0.69	0.69	$0.306s$
8	5.30	2.72	0.66	0.64	$2.625s$
9	5.92	2.83	0.64	0.59	$27.058s$

Tabela 12: Prosečne vrednosti statistika za $n < 10$

Izračunate vrednosti omogućuju procenu konstanti u asimptotskim izrazima za vrednosti širine (dužine najdužeg ciklusa) i visine (ukupnog broja ciklusa).

Funkcija `procena_x` (videti dodatak A.4) koristi `statistics_small_n` da prikupi prosečne vrednosti širine dijagrama, a zatim crta dijagram zavisnosti odstupanja ovih prosečnih vrednosti od linearne funkcije. Podsećanja radi, procena je $\hat{x} \approx c \cdot n$. Na početku se ispisuje lista odstupanja ($c = \frac{x}{n}$). Na dijagramu se može uočiti da se c približava vrednosti 0.66 kako n raste (ovo je samo neprecizna procena vrednosti konstante c, s obzirom na to da ovakva suma divergira).

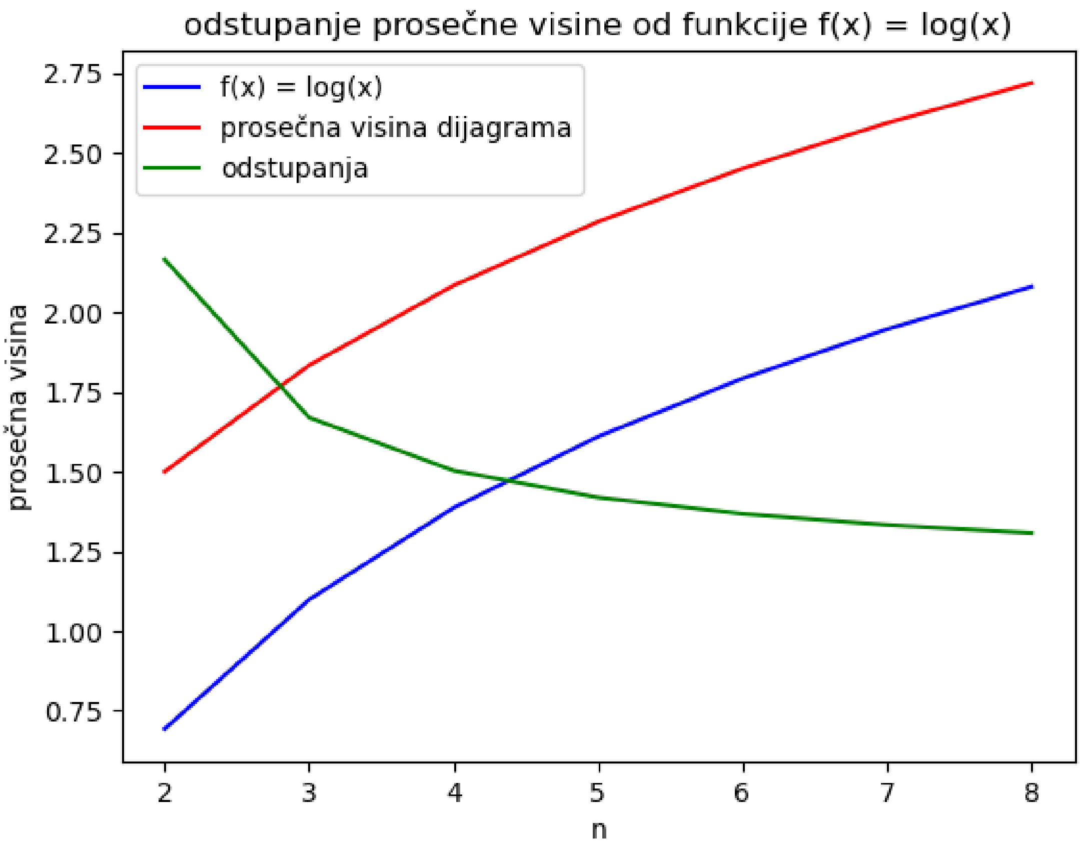
lista_odstupanja = [0.75, 0.72222, 0.69792, 0.685, 0.67384, 0.66789, 0.66203]



Slika 3.12: Odstupanje prosečne širine od funkcije $f(x) = x$

Funkcija `procena_y` (videti dodatak A.5) koristi `statistics_small_n` da prikupi prosečne vrednosti virine dijagrama, a zatim crta dijagram zavisnosti odstupanja ovih prosečnih vrednosti od funkcije $\log(n)$. Podsećanja radi, procena je $\hat{y} \approx c \cdot \log(n)$. Na početku se ispisuje lista odstupanja ($c = \frac{y}{\log(n)}$). Na dijagramu se može uočiti da se c približava vrednosti 1.3 kako n raste (ovo je samo neprecizna procena vrednosti konstante c, s obzirom na to da ovakva suma divergira).

$$\text{lista\_odstupanja} = [2.16404, 1.66877, 1.50281, 1.41872, 1.36737, 1.33246, 1.30701]$$



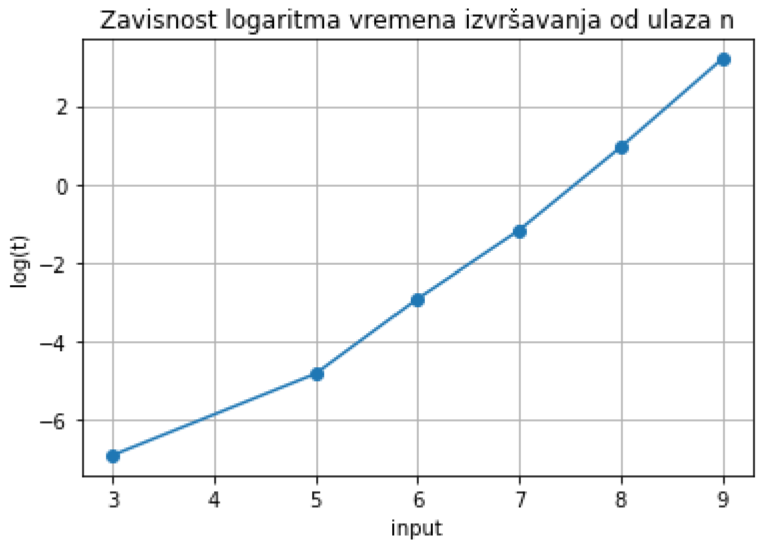
Slika 3.13: Odstupanje prosečne visine od funkcije $f(x) = log(x)$

3.1.1 Vreme izračunavanja

Za merenje vremena izračunavanja glavne funkcije koristi se funkcija `time_small_n` (videti dodatak A.6). Ova funkcija za ulaz ima broj n. Na početku uzima vreme početka izvršavanja, poziva funkciju koja računa statistike, po zavrsetku rada te funkcije, uzima vreme završetka. Povratna vrednost je razlika ova dva vremena, odnosno vreme u sekundama potrebno da se izvrši fukcija `statistics_small_n`. Funkcija `diagram_small_n` (videti dodatak A.7) iskorišćena je za crtanje grafika zavisnosti logaritma vremena izvršavanja od veličine ulaza n.

U funkciji se formira lista `ulazi`, koja sadrži nekoliko vrednosti koje odgovaraju opsegu za broj n. Zatim se u petlji poziva funkcija za računanje vremena t za svaku od vrednosti i logaritmi vremena $\log t$ se čuvaju u listi `vremena`. Elementi lista `ulazi` i `vremena` su neophodni parametri za crtanje dijagrama zavisnosti.

Na slici 3.1.1 je prikazan ispis ove funkcije:



Slika 3.1.1 Dijagram zavisnosti logaritma vremena od ulaza n

Zapaža se da je brzina rasta funkcije nešto veća od linearne, što je i očekivano, jer se za fiksirano n obrađuje $n!$ permmutacija.

3.2 Tačne statistike za $n \leq 72$

Za veće vrednosti n postaje zahtevno računati proseke svih $n!$ permutacija. Efikasnije je pri računanju prosečnih vrednosti vršiti sumiranje po svim particijama, vodeći računa o tome koliko permutacija odgovara jednom Jangovom dijagramu.

Glavni deo programa je funkcija `statistics_medium_n`. Ulaz je broj n, izlaz su prosečne vrednosti za širinu, visinu, punoću i asimetriju Jangovog dijagrama. Za sve particije (videti dodatak A.8) broja n određuje se njihov Jangov dijagram (funkcija koja konstruiše Jangov dijagram je ista kao u odeljku 3.1), množe se statistike (širina, visina, punoća i asimetrija) sa $N(D)$ i odvojeno sabiraju. Na kraju se četiri zbira dele sa $n!$, zbirom $N(D)$ po svim particijama (neke permutacije daju iste Jangove dijagrame - svaka statistika se množi brojem ponavljanja tog dijagrama).

```
def statistics_medium_n(n):
    x_sum = 0
    y_sum = 0
    lam_sum = 0
    asim_sum = 0
    particije = SveParticije(n)

    for particija in particije:
        (x, y, lam, asim, De, Ne) =
        Jangov_dijagram(particija, n)
        x_sum += x*Ne
        y_sum += y*Ne
        lam_sum += lam*Ne
```

```
        asim_sum += asim*Ne
    x_avg = x_sum/faktorijel(n)
    y_avg = y_sum/faktorijel(n)
    lam_avg = lam_sum/faktorijel(n)
    asim_avg = asim_sum/faktorijel(n)

    return x_avg, y_avg,lam_avg,asim_avg
```

Ova funkcija je dovoljno efikasna da radi za dvocifrene vrednosti broja n. Za ispis se koristi funkcija `print_statistics_medium_n` (videti dodatak A.9). Na primer, za $n = 37$ dobija se sledeći rezultat:

```
print_statistics_medium_n(37)
```

```
Prosečne vrednosti su:
širina: 23.41, dužina:  4.20, punoća: 0.46, asimetrija: 0.22
```

U sledećoj tabeli prikazane su prosečne vrednosti statistika za sve vrednosti $n \le 72$, kao i vremena izvršavanja u zavisnosti od broja n. Ova tabela se razlikuje od tabele iz odeljka 2.2 po tome što su ovde prosečne vrednosti izračunate po svim particijama broja n, dok su u tabeli 11 računate za slučajnu permutaciju reda n.

n	$\hat{x}$	$\hat{y}$	$\hat{\lambda}$	$\hat{\mu}$	vreme
13	8.42	3.18	0.58	0.46	$0.007s$
14	9.05	3.25	0.57	0.44	$0.009s$
15	9.67	3.32	0.56	0.42	$0.013s$
16	10.30	3.38	0.55	0.40	$0.018s$
17	10.92	3.44	0.55	0.38	$0.024s$
18	11.55	3.50	0.54	0.37	$0.033s$
19	12.17	3.55	0.53	0.36	$0.045s$
20	12.79	3.60	0.53	0.34	$0.062s$

n	$\hat{x}$	$\hat{y}$	$\hat{\lambda}$	$\hat{\mu}$	vreme
21	13.42	3.65	0.52	0.33	$0.063s$
22	14.04	3.69	0.51	0.32	$0.106s$
23	14.67	3.73	0.51	0.31	$0.110s$
24	15.29	3.78	0.50	0.30	$0.157s$
25	15.92	3.82	0.50	0.29	$0.198s$
26	16.54	3.85	0.49	0.28	$0.248s$
27	17.17	3.89	0.49	0.27	$0.326s$
28	17.79	3.93	0.49	0.27	$0.394s$
29	18.42	3.96	0.48	0.26	$0.496s$
30	19.04	3.99	0.48	0.25	$0.617s$
31	19.66	4.03	0.47	0.25	$0.783s$
32	20.29	4.06	0.47	0.24	$0.966s$
33	20.91	4.09	0.47	0.24	$1.185s$
34	21.54	4.12	0.46	0.23	$1.483s$
35	22.16	4.15	0.46	0.23	$1.916s$
36	22.79	4.17	0.46	0.22	$2.217s$
37	23.41	4.20	0.46	0.22	$2.716s$
38	24.03	4.23	0.45	0.21	$3.351s$
39	24.66	4.25	0.45	0.21	$4.066s$

n	$\hat{x}$	$\hat{y}$	$\hat{\lambda}$	$\hat{\mu}$	vreme
40	25.28	4.28	0.45	0.20	$4.917s$
41	25.91	4.30	0.45	0.20	$6.082s$
42	26.53	4.33	0.44	0.20	$7.266s$
43	27.16	4.35	0.44	0.19	$8.783s$
44	27.78	4.37	0.44	0.19	$10.783s$
45	28.41	4.39	0.44	0.19	$12.887s$
46	29.03	4.42	0.43	0.18	$16.65s$
47	29.65	4.44	0.43	0.18	$24.123s$
48	30.28	4.46	0.43	0.18	$29.727s$
49	30.90	4.48	0.43	0.17	$34.688s$
50	31.53	4.50	0.43	0.17	$42.169s$
51	32.15	4.52	0.42	0.17	$44.568s$
52	32.78	4.54	0.42	0.17	$55.776s$
53	33.40	4.56	0.42	0.16	$52.948s$
54	34.02	4.58	0.42	0.16	$63.663s$
55	34.65	4.59	0.42	0.16	$74.947s$
56	35.27	4.61	0.42	0.16	$88.496s$
57	35.90	4.63	0.42	0.15	$117.678s$
58	36.52	4.65	0.41	0.15	$123.094s$

n	$\hat{x}$	$\hat{y}$	$\hat{\lambda}$	$\hat{\mu}$	vreme
59	37.15	4.66	0.41	0.15	$145.077s$
60	37.77	4.68	0.41	0.15	$170.478s$
61	38.40	4.70	0.41	0.15	$201.224s$
62	39.02	4.71	0.41	0.14	$235.623s$
63	39.64	4.73	0.41	0.14	$276.698s$
64	40.27	4.74	0.41	0.14	$322.469s$
65	40.89	4.76	0.40	0.14	$499.154s$
66	41.52	4.77	0.40	0.14	$448.046s$
67	42.14	4.79	0.40	0.14	$513.194s$
68	42.77	4.80	0.40	0.13	$603.988s$
69	43.39	4.82	0.40	0.13	$684.217s$
70	44.01	4.83	0.40	0.13	$811.495s$
71	44.64	4.85	0.40	0.13	$1023.756s$
72	45.26	4.86	0.40	0.13	$1367.227s$

Tabela 13: Prosečne vrednosti statistika za $n \leq 72$

Zanimljivo je uporediti rezultate rada funkcije `statistics_small_n`, koja proseke statistika računa po svim permutacijama reda n i funkcije `statistics_medium_n` za malo n, koja proseke statistika računa po svim particijama broja n. Na primer, ispis funkcije `statistics_medium_n` za $n = 9$:

```
print_statistics_medium_n(9)
```

```
Prosečne vrednosti su:
širina: 5.92, dužina:  2.83, punoća: 0.64, asimetrija: 0.59
```

Poređenjem sa ispisom funkcije `statistics_small_n` za $n = 9$, može se uočiti da daju iste rezultate.

```
print_statistics_small_n(9)
```

```
Prosečne vrednosti su:
širina: 5.92, dužina:  2.83, punoća: 0.64, asimetrija: 0.59
```

Prosečne vrednosti izračunate korišćenjem funkcije `statistics_small_n` mogu se videti u odeljku 3.1, vrednosti izračunate korišćenjem funkcije `statistics_medium_n` mogu se pogledati u tabeli koja sledi.

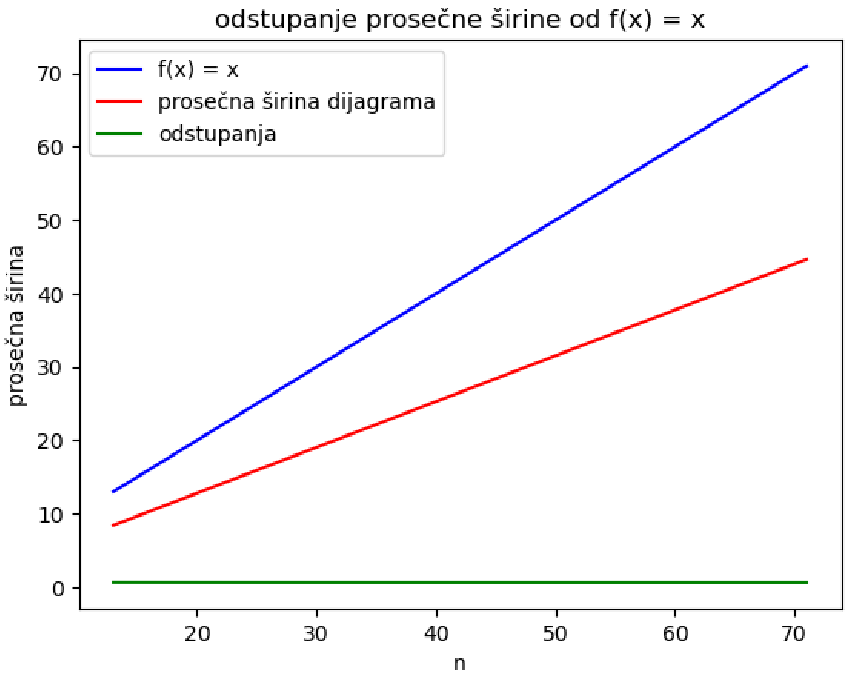
n	$\hat{x}$	$\hat{y}$	$\hat{\lambda}$	$\hat{\mu}$	vreme izvršavanja
4	2.79	2.08	0.81	0.95	$0.001s$
5	3.42	2.28	0.75	0.84	$0.001s$
6	4.04	2.45	0.72	0.76	$0.001s$
7	4.68	2.59	0.69	0.69	$0.001s$
8	5.30	2.72	0.66	0.64	$0.001s$
9	5.92	2.83	0.64	0.59	$0.002s$

Tabela 14: Rezultat izvršavanja funkcije *statistics_medium_n* za $n < 10$

Zaključuje se da obe funkcije imaju isti izlaz za svako n, ali je vreme izračunavanja proseka statistika po particijama znatno kraće.

Izračunate vrednosti omogućuju procenu konstanti u asimptotskim izrazima za vrednosti širine (dužine najdužeg ciklusa) i visine (ukupnog broja ciklusa).

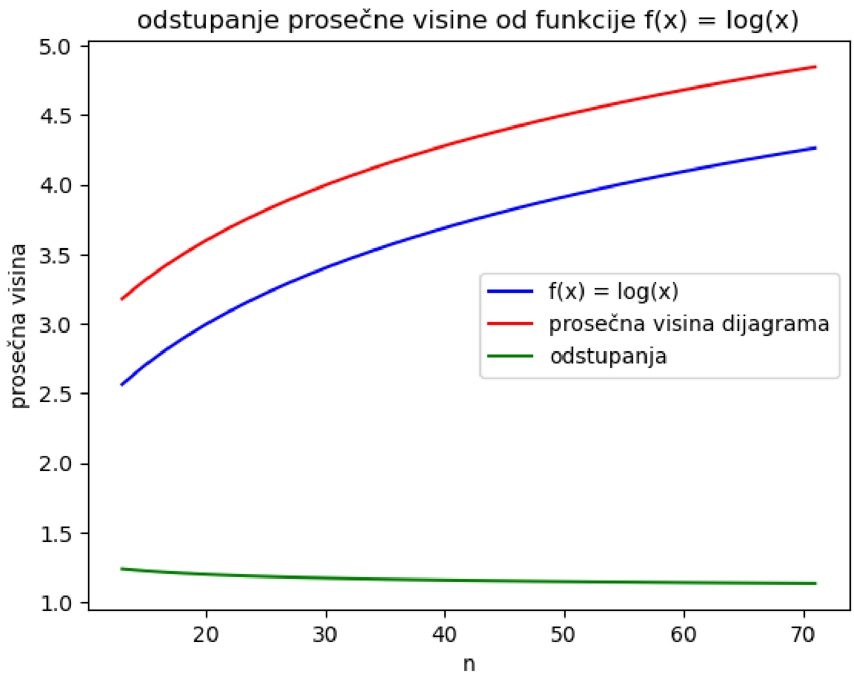
Funkcija `procena_x_medium` (dodatak A.17) koristi `statistics_medium_n` da prikupi prosečne vrednosti širine dijagrama, a zatim crta dijagram zavisnosti odstupanja ovih prosečnih vrednosti od linearne funkcije. Procena je $\hat{x} \approx c \cdot n$. Na dijagramu se može uočiti da se c približava vrednosti 0.62 kako n raste (ovo je samo neprecizna procena vrednosti konstante c, s obzirom na to da ovakva suma divergira):



Slika 3.1: Odstupanje prosečne širine od funkcije $f(x) = x$

Funkcija `procena_y_medium` (dodatak A.18) koristi `statistics_medium_n` da prikupi prosečne vrednosti virine dijagrama, a zatim crta dijagram zavisnosti odstupanja ovih prosečnih vrednosti od funkcije $\log(n)$. Podsećanja radi, procena je $\hat{y} \approx c \cdot \log(n)$. Na dijagramu (prikazan je na slici 3.2) se može uočiti da se c

približava vrednosti 1.14 kako n raste (ovo je samo neprecizna procena vrednosti konstante c, s obzirom na to da ovakva suma divergira).

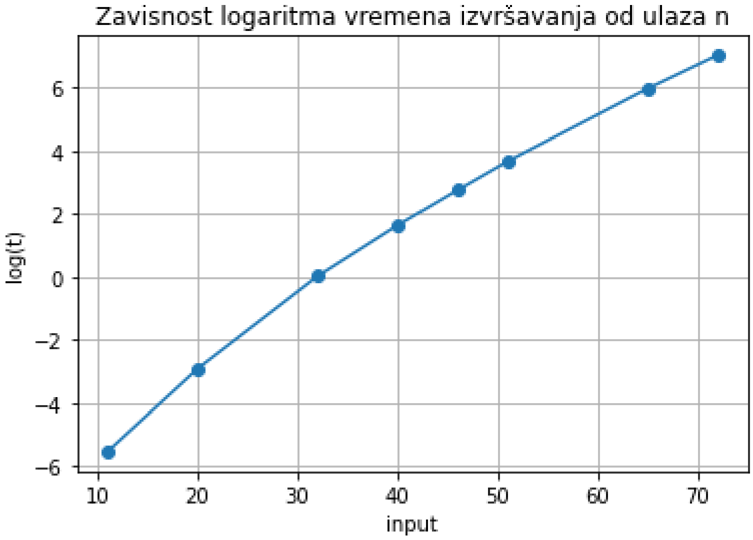


Slika 3.2: Odstupanje prosečne visine od funkcije $f(x) = log(x)$

3.2.1 Vreme izračunavanja

Za merenje vremena izračunavanja glavne funkcije koristi se funkcija `time_medium_n` (njen potpis pogledati u dodatku A.10), koja računa vreme potrebno da se izvrši funkcija `statistics_medium_n`:

Ova funkcija ima potpuno isti opis kao funkcija iz odeljka 3.1.1, koja radi isti posao za manje vrednosti n. Takođe, i funkcija za crtanje grafika je ista kao ona iz dela 3.1.1. Njen ispis u ovom slučaju je prikazan na slici 3.2.1



Slika 3.2.1 Dijagram zavisnosti logaritma vremena od ulaza n

Zapaža se da se vreme izvršavanja ponaša subeksponencijalno. Ova činjenica je u skladu sa asimptotskim izrazom za broj particija:

$$p(n) \sim \frac{1}{4n\sqrt{3}} e^{\pi\sqrt{\frac{2n}{3}}} \ (n \longleftrightarrow \infty)$$

(videti [7])

3.3 Ocena statistika metodom Monte Karlo

Kako vrednost broja n raste, postaje sve teže izvršiti potrebna izračunavanja. Ovaj odeljak posvećen je računanju prosečnih statistika za velike vrednosti broja $n > 100$. U odeljku 1 opisan je postupak koji se u ovoj glavi primenjuje korišćenjem Python koda.

Glavni deo programa je funkcija `statistics_large_n` koja za ulazne parametre uzima prirodan broj n, kao i sve prethodne funkcije za računanje statistika, i prirodan broj k koji predstavlja broj permutacija za generisanje. Na početku u petlji se k puta poziva funkcija `slucajna_permutacija`, koja generiše slučajnu permutaciju reda n. Za svaku takvu permutaciju se poziva funkcija `particionisanje`, koja od permutacije pravi particiju prebrojavajući cikluse. Zatim se za svaku tako generisanu particiju poziva funkcija koja konstruiše Jangov dijagram i određuje njegove karakteristike, slično funkciji iz odeljka 3.1. U okviru petlje vrši se sumiranje vrednosti, dok se na samom kraju svaka suma deli brojem k. Funkcija kao povratne vrednosti ima prosečne statistike po generisanim permutacijama.

```
def statistics_large_n(n, k):
x_sum = 0
y_sum = 0
lam_sum = 0
asim_sum = 0

for i in range (0,k):
    perm = slucajna_permutacija(n)
    part = particionisanje(perm)
    (x, y, lam, asim, De) = Jangov_dijagram(part, n)
    x_sum += x
```

```
        y_sum += y
        lam_sum += lam
        asim_sum += asim
    x_avg = x_sum/k
    y_avg = y_sum/k
    lam_avg = lam_sum/k
    asim_avg = asim_sum/k

    return x_avg, y_avg,lam_avg, asim_avg
```

Prethodno opisana funkcija poziva funkciju `slucajna_permutacija` (čiji kod se može videti u dodatku A.11) koja za ulaz ima prirodan broj n, formira niz $\{0, 1, ...n - 1\}$, a zatim koristi Fišer Jetsovo mešanje (videti 2.2) za generisanje slučajne permutacije.

Glavna funkcija poziva još i deo kôda za `particionisanje` koji za ulaznu permutaciju, u listu ciklusi ubacuje sve cikluse koje obrazuje data permutacija. Zatim u petlji za svaki ciklus uzima njegovu dužinu i taj broj dodaje u listu particije. Ovaj deo kôda može se videti u dodatku A.12.

U funkciji `particionisanje` poziva se funkcija `ciklusi` koja prima jednu permutaciju skupa veličine n. Prolazi kroz permutaciju i generiše cikluse, dodaje njihove dužine u listu, koja je povratna vrednost funkcije (videti dodatak A.13).

U nastavku je prikazan rad ovog dela programa. Za ispis se koristi funkcija `print_statistics_large_n` (pogledati dodatak A.14).
Na primer, za $n = 2^9$ i $k = 100$ dobija se:

```
print_statistics_large_n(512, 100)
```

```
Prosečne vrednosti su:
širina: 283.14, dužina:  6.58, punoća: 0.62, asimetrija: 0.06
```

Sledeća tabela prikazuje rezultate izvršavanja funkcije `statistics_large_n` za $n = 2^i$ i $k = 10$, kao i vreme izvršavanja:

n	$\hat{x}$	$\hat{y}$	$\hat{\lambda}$	$\hat{\mu}$	vreme
2^3	6.10	1.90	0.99	0.569	$0.001s$
2^4	7.40	3.30	1.15	0.883	$0.002s$
2^5	9.80	4.90	1.53	1.439	$0.002s$
2^6	25.80	5.40	1.15	0.586	$0.003s$
2^7	59.90	6.00	0.71	0.230	$0.003s$
2^8	128.40	5.80	0.94	0.264	$0.005s$
2^9	235.30	8.10	0.38	0.048	$0.012s$
2^{10}	466.00	7.50	0.60	0.032	$0.020s$
2^{11}	1033.90	7.90	1.53	0.034	$0.044s$
2^{12}	2224.40	8.50	0.58	0.006	$0.147s$
2^{13}	3477.70	9.50	0.75	0.006	$0.182s$
2^{14}	8331.80	9.70	0.46	0.002	$0.357s$
2^{15}	14427.90	11.20	0.86	0.003	$0.741s$
2^{16}	31672.50	10.80	0.32	0.000	$1.534s$
2^{17}	57778.70	11.30	1.64	0.003	$3.508s$
2^{18}	131124.90	15.20	0.41	0.000	$7.759s$

Tabela 15: Ispis funkcije *statistics_ large_ n* za $n = 2^i$ i $k = 10$ i vreme izvršavanja

U sledećoj tabeli prikazani su rezultati izvršavanja funkcije `statistics_large_n` za $n = 2^i$ i $k = 100$, kao i vreme izvršavanja:

n	$\hat{x}$	$\hat{y}$	$\hat{\lambda}$	$\hat{\mu}$	vreme
2^3	4.57	2.74	0.97	1.109	$0.002s$
2^4	8.55	3.37	1.13	0.940	$0.004s$
2^5	15.45	4.41	0.89	0.633	$0.007s$
2^6	32.30	4.56	0.90	0.270	$0.012s$
2^7	63.32	5.51	0.86	0.222	$0.029s$
2^8	127.41	5.98	1.36	0.312	$0.055s$
2^9	244.28	6.74	0.97	0.201	$0.124s$
2^{10}	515.46	7.77	0.98	0.073	$0.215s$
2^{11}	971.01	8.36	0.84	0.024	$0.401s$
2^{12}	2060.33	8.92	0.42	0.009	$0.797s$
2^{13}	3849.56	9.47	1.30	0.017	$1.625s$
2^{14}	7672.28	15.58	0.96	0.008	$3.276s$
2^{15}	15429.76	10.99	0.64	0.002	$6.904s$
2^{16}	33165.19	11.82	0.65	0.001	$15.016s$
2^{17}	66039.84	12.17	0.39	0.000	$34.284s$
2^{18}	130567.12	13.27	3.00	0.005	$78.144s$

Tabela 16: Ispis funkcije za $n = 2^i$ i $k = 100$ i vreme izvršavanja

Napomena: Ispisi ove funkcije se menjaju pri svakom pozivu, jer svaki put program generiše nove permutacije. U odeljku 3.3.2 analizirana je procena tačnosti.

3.3.1 Vreme izračunavanja

Za merenje vremena izračunavanja glavne funkcije koristi se funkcija `time_large_n`:

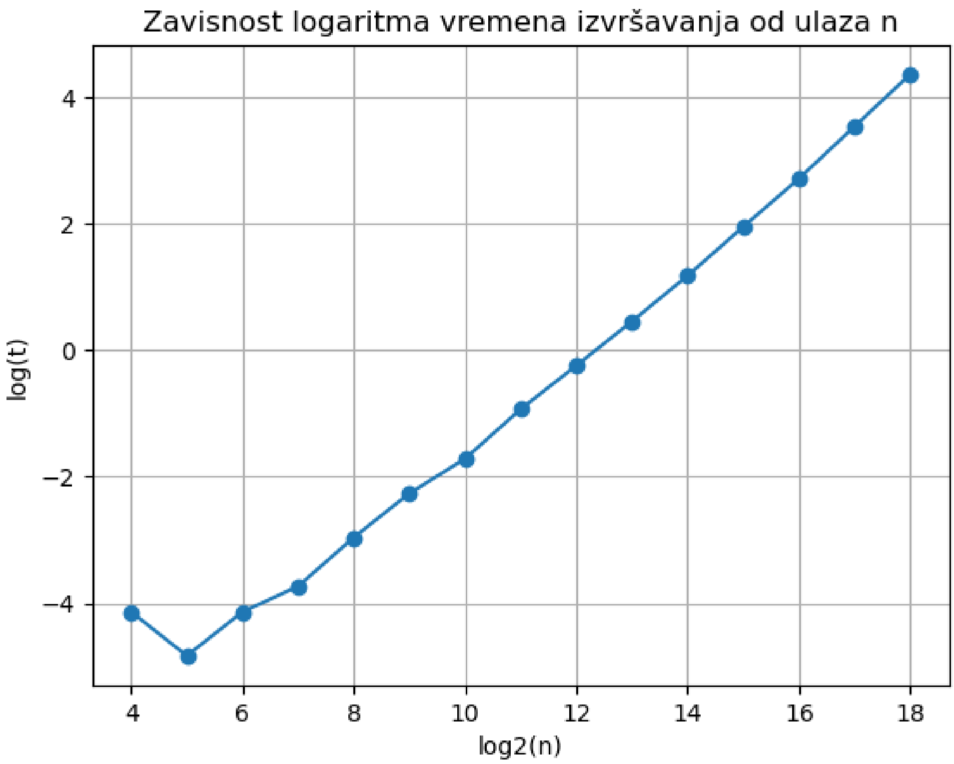
```
import time
def time_large_n(n, k):

    start_time = time.time()
    (x, y, lam, mi) = statistics_large_n(n,k)
    end_time = time.time()

    execution_time = end_time - start_time
    return execution_time
```

Ova funkcija ima isti opis kao funkcija iz odeljka 3.1.1, koja radi isti posao za manje vrednosti n. Razlikuju se po tome što funkcija `time_large_n` kao ulazni parametar ima i broj k. I funkcija `diagram_large_n` ima vrlo sličan opis odgovarajućoj funkciji iz istog odeljka. Razlika je što funkcija `diagram_large_n` ima ulazni parametar k (videti dodatak A.15).

Na slici 3.3.1 prikazan je dijagram zavisnosti logaritma vremena izvršavanja od n za $k = 100$.



Slika 3.3.1: Zavisnost logaritma vremena izvršavanja od ulaza n

Na dijagramu se može primetiti brzina rasta vremena, na koju najviše utiče deo programa za generisanje permutacija. Za $n = 2^{11}$ prikazano je poređenje vremena za generisanje 100 slučajnih permutacija i vremena potrebnog za računjanje prosečnih statistika:

```
time_large_n(4096, 100)
0.8462212085723877
```

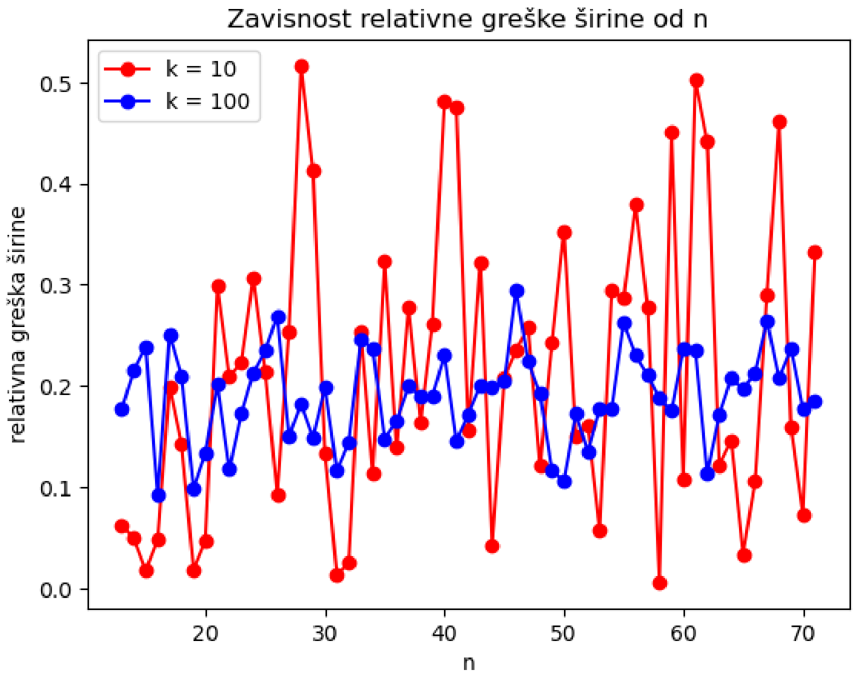
```
time_permutations(4096, 100)
0.7059564590454102
```

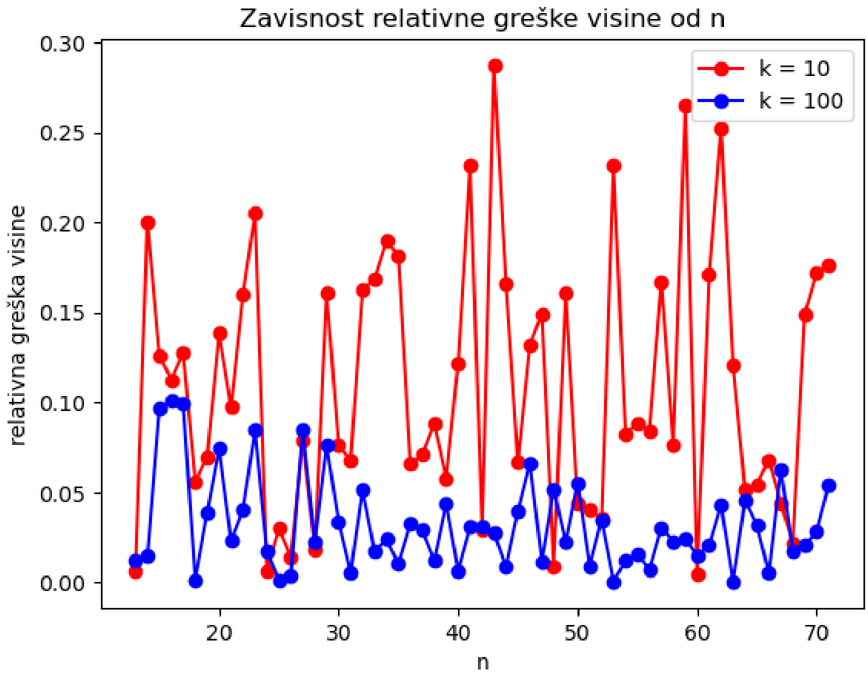
Za računanje vremena potrebnog za generisanje k slučajnih permutacija korišćena je funkcija `time_permutations` čiji potpis se može videti u dodatku A.21.

3.3.2 Procena tačnosti približnih vrednosti statistika za $n \leq 72$

U ovom delu rada prikazana je procena tačnosti približnih vrednosti statistika koje su izračunate na način opisan u odeljku 3.3 u odnosu na tačne prosečne vrednosti iz odeljka 3.2. Za ovu svrhu koristi se funkcija `procena_tacnosti_do_100`, koja formira listu vrednosti n, zatim poziva funkcije za izračunavanje tačne i približne vrednosti prosečnih statistika (za $k = 10$ i $k = 100$). Nakon toga računa relativnu grešku za svaku karakteristiku i crta dijagram zavisnosti greške od veličine n. Ovaj deo kôda može se pogledati u dodatku A.16.

U nastavku je su prikazani dijagrami zavisnosti:





Posmatrajući prethodne dijagrame može se uočiti da relativna greška procene širine i visine više osciluje za $k = 10$. Ovo sledi iz činjenice da se za različite vrednosti n svaki put generišu slučajne permutacije. Nekada generisane permutacije formiraju particije čiji dijagrami imaju statistike bliske prosečnim vrednostima, dok nekada generisane permutacije formiraju particije čiji dijagrami imaju statistike koje prilično odstupaju od prosečnih vrednosti. Za $k = 100$ oscilacije su manje, jer pri generisanju više permutacija veće su šanse da one formiraju particije čiji Jangovi dijagrami imaju statistike bliske prosečnim vrednostima. Izostavljeni su dijagami procene greške punoće i asimetrije, jer u nekim pozivima, generišu se slučajne permutacija koje formiraju particije čiji su Jangovi dijagrami prilično asimetrični (slično važi i za punoću), odakle sledi da prosek ovakvih vrednosti mnogo odstupa od tačne prosečne vrednosti (računate pomoču funkcije `statistic_medium_n`), stoga je vrednost relativne greške neprirodno velika.

4 Zaključak

U ovom radu predstavljen je koncept Jangovih dijagrama i njihova veza sa particijama, kao i veza permutacija i particija. Prikazano je kako se vizuelno mogu prikazati elementi kombinatorike.

Tok rada mogao bi se predstaviti grafički na sledeći način:

$$\begin{gathered}\text{prirodan broj } n\\ \downarrow\\ \text{skup } \{0,1,2,...,n-1\}\\ \downarrow\\ \text{permutacija skupa}\\ \downarrow\\ \text{generisanje ciklusa}\\ \downarrow\\ \text{particija}\\ \downarrow\\ \text{Jangov dijagram}\\ \downarrow\\ \text{statistike}\\ \downarrow\\ \text{prosečne vrednosti statistika}\end{gathered}$$

Na početku izložen je teorijski uvod i opisani osnovni pojmovi, koji su korišćeni u radu, kao što su: permutacije, particije, Jangovi dijagrami, statistike, simetrične grupe, metod Monte Karlo.

Deo rada posvećen je predstavljenju teorijskih izračunavanja statistika. Dati su rezultati izračunavanja za male vrednosti n (videti 2.1), dok su u odeljku 2.2 prikazani rezultati za nešto veće vrednosti broja n.

Prikazana je eksperimentalna provera rezultata. U jednom delu progama prosek statistika se računa po svim permutacijama (za $n \leq 9$), dok drugi deo programa računa procenu statistika (za $n \leq 72$) po svim particijama. Treći deo predstavlja segment programa koji računa procenu statistika po slučajno generisanim permutacijama. Predstavljena je zavisnost logaritma vremena izvršavanja ovih delova programa u zavisnosti od veličine ulaza n, kao i procena greške pri računanju sa slučajnim permutacijama.

Ispitujući prosečne vrednosti statistika može se primetiti da vrednosti konstanti u asimptotskim izrazima za statistike Jangovih dijagrama konvergiraju kako se veličina dijagrama povećava. Veza između simetričnih grupa i Jangovih dijagrama može biti izuzetno zanimljiva u daljem istraživanju. Mogući pravac za dalji rad bi mogao biti sumiranje po slučajno izabranim particijama, što bi dalje povećalo opseg dostižnih vrednosti za n.

Literatura

[1] V. Arnold, *Experimental Mathematics* (Translated from russian). (2015) A co-publication of the AMS and Mathematical Sciences Research Institute

[2] B. Sagan, *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*, 2nd edition. (2001) Springer-Verlag, New York

[3] G. Andrews, *The Theory of Partitions* (1998) Cambridge University Press

[4] B. Baumslag, B. Chandler, Schaum's *Outline of Theory and problems of Group Theory* (1968) McGRAW HILL New York

[5] D. Knuth *Seminumerical algorithms. The Art of Computer Programming.* (1969) Addison–Wesley

[6] D. Tomić *Algoritmi za generisanje particija* (2020) Matematički fakultet, Univerzitet u Beogradu

[7] H. Wilf *Lectures on Integer Partitions* (2000) University of Pennsylvania

[8] L. B. Koralov, Y. G. Sinai Y.G., *Theory of Probability and Random Processes* (2007) Springer

A Dodatak: Kôd

U prilogu se nalaze sve funkcije korišćene za izračunavanja u glavi 3, dok su u radu predstavljene samo osnovne, koje se pozivaju na ove iz dodatka.

A.1 Funkcija koja generiše particije za sve permutacije skupa veličine n

```
def particije(n):
    permutacije = sve_permutacije(n)
    particije = []

    for permutacija in permutacije:
        particija = particionisanje(permutacija)
        particija.sort(reverse = True)
        particije.append(particija)

    return particije
```

A.2 Kôd koji generiše sve permutacije skupa veličine n

```
from itertools import permutations
def sve_permutacije(n):
    lista = [x for x in range (0,n)]
    permutacije = []
    perms = permutations(lista)
    for perm in perms:
        permutacija = list(perm)
        permutacije.append(permutacija)
    return permutacije
```

A.3 Ispis vrednosti prosečnih statistika za malo n

```
def print_statistics_small_n(n):
    x_avg, y_avg,lam_avg, asim_avg = statistics_small_n(n)
    poruka = "Proscne vrednosti dijagrama su: sirina: {:.2f},
    duzina:  {:.2f}, punoca: {:.2f}, asimetrija: {:.2f}"
    print(poruka.format(x_avg, y_avg,lam_avg, asim_avg))
```

A.4 Odstupanje prosečne vrednosti sirine od linearne funkcije za malo n

```
import math
import matplotlib.pyplot as plt
def procena_x():
    niz_x = []
    niz_n = [x for x in range(2, 9)]
    niz_odstupanja = []

    for n in niz_n:
```

```
        (x_a, y_a, lam_a, asim_a) = statistics_small_n(n)
        niz_x.append(x_a)
        odstupanje = round((x_a / n),6)
        niz_odstupanja.append(odstupanje)

    print(niz_odstupanja)
    fig, ax = plt.subplots()
    ax.plot(niz_n, niz_n, color='blue',
    label='f(x) = x')
    ax.plot(niz_n, niz_x, color='red',
    label='prosecna sirina dijagrama')
    ax.plot(niz_n, niz_odstupanja, color = 'green',
    label = 'odstupanja')

    ax.set_xlabel('n')
    ax.set_ylabel('prosecna sirina')
    ax.set_title('odstupanje prosecne sirine od f(x) = x')
    ax.legend()

    plt.show()
```

A.5 Odstupanje prosečne vrednosti visine od logaritamske funkcije za malo n

```
import matplotlib.pyplot as plt
def procena_y():
    niz_y = []
    niz_n = [x for x in range(2, 9)]
    niz_ln = [math.log(x) for x in range(2, 9)]
    niz_odstupanja = []

    for n in niz_n:
        (x_a, y_a, lam_a, asim_a) = statistics_small_n(n)
        niz_y.append(y_a)
```

```
        odstupanje = round((y_a / math.log(n)),6)
        niz_odstupanja.append(odstupanje)

    print(niz_odstupanja)
    fig, ax = plt.subplots()
    ax.plot(niz_n, niz_ln, color='blue',
    label='f(x) = log(x)')
    ax.plot(niz_n, niz_y, color='red',
    label='prosecna visina dijagrama')
    ax.plot(niz_n, niz_odstupanja, color = 'green',
    label = 'odstupanja')

    ax.set_xlabel('n')
    ax.set_ylabel('prosecna visina')
    ax.set_title('odstupanje prosecne visine od
    funkcije f(x) = log(x)')
    ax.legend()

    plt.show()
```

A.6 Funkcija za merenje vremena izvršavanja funkcije *statistics_small_n*

```
import time
def time_small_n(n):

    start_time = time.time()
    (x, y, lam, mi) = statistics_small_n(n)
    end_time = time.time()

    execution_time = end_time - start_time
    return execution_time
```

A.7 Crtanje dijagrama zavisnosti logaritma vremena izvršavanja funkcije *statistics_small_n* od veličine ulaza

```
import time
import math
import matplotlib.pyplot as plt

def diagram_small_n():
    ulaz = [3, 5, 6, 7, 8, 9]
    vremena = []

    for param in ulaz:
        vreme = time_small_n(param)
        vreme = math.log(vreme)
        vremena.append(vreme)

    plt.plot(ulaz, vremena, marker='o')
    plt.xlabel('input')
    plt.ylabel('log(t)')
    plt.title('Zavisnost logaritma vremena izvrsavanja od
    ulaza n')
    plt.grid(True)
    plt.show()
```

A.8 Kôd za pronalaženje svih particija broja n

```
    def stampaj_niz(niz, n):
    for i in range(0, n):
        print(niz[i], end = " ")
    print()

def dodaj_particiju(adresa, niz, n):
    niz = niz[0:n]
    adresa.append(niz)
```

```
def SveParticije(n):
    particija = [0] * n
    k = 0
    particija[k] = n
    particije = []

    while True:
        dodaj_particiju(particije, particija, k+1)
        tmp = 0
        while k >= 0 and particija[k] == 1:
            tmp += particija[k]
            k -= 1

        if k < 0:
            return particije

        particija[k] -= 1
        tmp += 1
        while tmp > particija[k]:
            particija[k + 1] = particija[k]
            tmp = tmp - particija[k]
            k += 1

        particija[k + 1] = tmp
        k += 1
```

A.9 Funkcija za ispis vrednosti funkcije *statisttics_medium_n*

```
    def print_statistics_medium_n(n):
    x_avg, y_avg,lam_avg, asim_avg = statistics_medium_n(n)
    poruka = "Prosecne vrednosti dijagrama su:
    sirina je: {:.2f}, duzina:  {:.2f},
    punoca: {:.2f}, asimetrija: {:.2f}"
```

```
    print(poruka.format(x_avg, y_avg,lam_avg, asim_avg))
```

A.10 Funkcija za merenje vremena izvršavanja funkcije *statistics_medium_n*

```
import time
def time_medium_n(n):
    start_time = time.time()
    (x, y, lam, mi) = statistics_medium_n(n)
    end_time = time.time()
    execution_time = end_time - start_time
    return execution_time
```

A.11 Deo kôda koji generiše slučajnu permutaciju reda n

```
from numpy import random
import numpy as np
def slucajna_permutacija(n):
    lista = [x for x in range(0, n)]
    perm = np.random.permutation(lista)
    permutacija = perm.tolist()
    return permutacija
```

A.12 Funkcija koja od permutacije formira particiju

```
def particionisanje(permutacija):

cikls = ciklusi(permutacija)
particija = []

for ciklus in cikls:
    duzina = len(ciklus)
    particija.append(duzina)
return particija
```

A.13 Funkcija koja vraća listu ciklusa

```
from sympy.combinatorics.partitions import Partition
from sympy.combinatorics.permutations import Permutation
def ciklusi(permutacija):
    perm = Permutation(permutacija)
    return perm.full_cyclic_form
```

A.14 Funkcija za ispis vrednosti funkcije *statistcs_ medium_ n*

```
def print_statistics_large_n(n, k):
    x_avg, y_avg,lam_avg, asim_avg = statistics_large_n(n, k)
    poruka = "Prosecne vrednosti dijagrama su:
    sirina: {:.2f}, duzina: {:.2f},
    punoca: {:.2f}, asimetrija: {:.2f}"
    print(poruka.format(x_avg, y_avg,lam_avg, asim_avg))
```

A.15 Funkcija za crtanje dijagrama zavisnosti logaritma vremena izvršavanja funkcije *statistics_ large_ n*

```
import time
import matplotlib.pyplot as plt

def diagram_large_n(k):
    ulaz = [16, 64, 128, 256, 512, 1024, 2048, 4096, 8192,
    16384, 131072]
    vremena = []

    for param in ulaz:
        vreme = time_large_n(param,k)
        vreme = math.log(vreme)
        vremena.append(vreme)

    plt.plot(ulaz, vremena, marker='o')
    plt.xlabel('input')
    plt.ylabel('log(t)')
    plt.title('Zavisnost logaritma vremena izvrsavanja od ulaza n')
    plt.grid(True)
    plt.show()
```

A.16 Funkcija koja crta dijagram relativne greške približnih vrednosti statistika

```
def procena_tacnosti_do_100():
    lista = [x for x in range(13,72)]

    greske_x = []
    greske_x1 = []
    greske_y = []
    greske_y1 = []


    for n in lista:
        (tacno_x, tacno_y, tacno_lam, tacno_mi) =
        statistics_medium_n(n)
```

```
        (priblizno_x, priblizno_y, priblizno_lam,
        priblizno_mi) =  statistics_large_n(n, 10)

        (priblizno_x1, priblizno_y1, priblizno_lam1,
        priblizno_mi1) = statistics_large_n(n, 100)

        rel_greska_x = abs(priblizno_x - tacno_x)/tacno_x
        rel_greska_y = abs(priblizno_y - tacno_y)/tacno_y

        rel_greska_x1 = abs(priblizno_x1 - tacno_x)/tacno_x
        rel_greska_y1 = abs(priblizno_y1 - tacno_y)/tacno_y

        greske_x.append(rel_greska_x)
        greske_y.append(rel_greska_y)

        greske_x1.append(rel_greska_x1)
        greske_y1.append(rel_greska_y1)


    print("Dijagrami zavisnosti relativne greske statistika su:")

    fig, ax = plt.subplots()
    ax.plot(lista, greske_x, color = 'red',
    label = 'k = 10', marker = 'o')
    ax.plot(lista, greske_x1, color = 'blue',
    label = 'k = 100', marker = 'o')
    ax.set_xlabel('n')
    ax.set_ylabel('relativna greska sirine')
    ax.set_title('Zavisnost relativne greske sirine od n')
    ax.legend()
    plt.show()

    fig, ax = plt.subplots()
    ax.plot(lista, greske_y, color = 'red',
    label = 'k = 10', marker = 'o')
    ax.plot(lista, greske_y1, color = 'blue',
    label = 'k = 100', marker = 'o')
```

```
    ax.set_xlabel('n')
    ax.set_ylabel('relativna greska visine')
    ax.set_title('Zavisnost relativne greske visine od n')
    ax.legend()
    plt.show()
```

A.17 Odstupanje prosečne vrednosti sirine od linearne funkcije za nešto veće n

```
import math
import matplotlib.pyplot as plt
def procena_x_medium():
    niz_x = []
    niz_n = [x for x in range(13, 72)]
    niz_odstupanja = []

    for n in niz_n:
        print(n)
        (x_a, y_a, lam_a, asim_a) = statistics_medium_n(n)
        niz_x.append(x_a)
        odstupanje = round((x_a / n),6)
        niz_odstupanja.append(odstupanje)

    fig, ax = plt.subplots()
    ax.plot(niz_n, niz_n, color='blue',
    label='f(x) = x')
    ax.plot(niz_n, niz_x, color='red',
    label='prosecna sirina dijagrama')
    ax.plot(niz_n, niz_odstupanja, color = 'green',
    label = 'odstupanja')

    ax.set_xlabel('n')
    ax.set_ylabel('proscna sirina')
    ax.set_title('odstupanje prosecne
    sirine od f(x) = x')
    ax.legend()
```

```
    plt.show()
```

A.18 Odstupanje prosečne vrednosti visine od logaritamske funkcije za nešto veće n

```
import math
import matplotlib.pyplot as plt
def procena_y_medium():
    niz_y = []
    niz_n = [x for x in range(13, 72)]
    niz_ln = [math.log(x) for x in niz_n]
    niz_odstupanja = []

    for n in niz_n:
        print(n)
        (x_a, y_a, lam_a, asim_a) = statistics_medium_n(n)
        niz_y.append(y_a)
        odstupanje = round((y_a / math.log(n)),6)
        niz_odstupanja.append(odstupanje)

    fig, ax = plt.subplots()
    ax.plot(niz_n, niz_ln, color='blue',
    label='f(x) = log(x)')
    ax.plot(niz_n, niz_y, color='red',
    label='prosecna visina dijagrama')
    ax.plot(niz_n, niz_odstupanja, color = 'green',
    label = 'odstupanja')

    ax.set_xlabel('n')
    ax.set_ylabel('prosecna visina')
    ax.set_title('odstupanje prosecne visine od
    funkcije f(x) = log(x)')

    ax.legend()
```

```
    plt.show()
```

A.19 Funkcija za ispis Jangovog dijagrama D

```
def D(particija):
    D = ""
    exp = 1

    if len(particija) == 1:
        D = D + str(particija[0])

    for i in range(0, len(particija) - 1):
        if particija[i] == particija[i+1]:
            exp += 1
            if i == len(particija) - 2:
                if exp == 1:
                    D = D + str(particija[i]) + "*"
                else:
                    D = D + str(particija[i]) + "^" + str(exp)

        else:
            if exp == 1:
                D = D + str(particija[i]) + "*"
            else:
                D = D + str(particija[i]) + "^"
                + str(exp) + "*"
                exp = 1

    if particija[len(particija) - 2] !=
    particija[len(particija)-1]:
        exp = 1
        D = D + str(particija[len(particija)-1])

    return D
```

A.20 Funkcija koja računa $N(D)$

```
import math

def N(particija):
    n = 0
    for elem in particija:
        n += elem

    indeks = faktorijel(n)
    proizvod = 1

    for elem in particija:
        m = particija.count(elem)
        proizvod *= (math.pow(elem,m))*faktorijel(m)
        particija = [x for x in particija if x != elem]

    indeks /= proizvod
    indeks = round(indeks)

    return indeks
```

A.21 Funkcija za računanje vremena potrebnog za generisanje k slučajnih permutacija reda n

```
import time
def time_permutations(n, k):
    vreme = 0
    for i in range (0, k):
        start = time.time()
        perm = slucajna_permutacija(n)
        end = time.time()
        vreme_p = round( end - start, 17)
        vreme = vreme + vreme_p
```

```
    return vreme
```