УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Тијана Никчевић

ПРЕБРОЈАВАЊЕ ГРАФИЧКИХ НИЗОВА

мастер рад

Београд, 2021.

Ментор:

др Миодраг Живковић, редовни професор
Универзитет у Београду, Математички факултет

Чланови комисије:

др Филип Марић, ванредни професор
Универзитет у Београду, Математички факултет

др Весна Маринковић, доцент
Универзитет у Београду, Математички факултет

Датум одбране:

Наслов мастер рада: Пребројавање графичких низова

Резиме: Графички низ је нерастући низ $d = (d_1, d_2, \ldots, d_n)$ за који постоји неусмерени граф $G = (V, E)$ такав да је скуп степенова његових чворова $\{d_1, d_2, \ldots, d_n\}$. Постоје различити начини да се провери да ли је низ графички. Према теореми Хавела и Хакимија, за $n \geq 3$, нерастући низ $d = (d_1, d_2, \ldots, d_n)$ је графички ако и само ако је низ $d' = (d_2 - 1, d_3 - 1, \ldots, d_{d_1} - 1, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n)$ графички. Према теореми Ердеша и Галаиа, нерастући низ $d = (d_1, d_2, \ldots, d_n)$ је графички ако и само ако је сума његових елемената парна и ако је неједнакост $\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} \min\{k, d_i\}$ тачна за свако k, $1 \leq k < n$. На основу теореме Хавела и Хакимија и теореме Ердеша и Галаиа развијени су различити алгоритми којима се проверава да ли је низ графички. Пребројавање графичких низова може да се уради излиставањем свих нерастућих низова и провером за сваки од њих да ли је графички. Овакав поступак има експоненцијалну сложеност. Ванг је 2019. представио алгоритам за пребројавање графичких низова који не излистава све нерастуће низове, већ рачуна број графичких низова коришћењем формула и има полиномијалну сложеност. У овом раду размотрени су и експериментално упоређени ови алгоритми.

Кључне речи: графички низ, графичка партиција, пребројавање, прости графови

Садржај

1. Увод

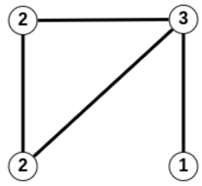
Графички низ је нерастући низ чији су елементи степени чворова неког графа. Графички низови су значајни за све проблеме који се могу моделовати графовима, а код којих је познат само број чворова графа и степен сваког чвора у графу. Захваљујући томе су нашли и практичну примену, нпр. код приказа хемијских структура. Различите структуре истог хемијског једињења представљају различите графове који имају исти графички низ [6]. Графички низови користе се и за моделовање мрежа у области комуникација, ланаца исхране и преносивих болести [11]. Такође имају примену и у спорту, нпр. код проблема везаних за моделовање турнира [10].

У одељку 1.1 дефинисани су основни појмови који се користе у овом раду, док је у одељку 1.2 појашњен проблем пребројавања графичких низова и дат преглед осталих поглавља овог рада.

1.1 Основни појмови

Граф $G = (V, E)$ је уређени пар скупова V и E такав да су елементи скупа E парови елемената из скупа V. Елементи скупа V називају се чворови, док се елементи скупа E називају гране. Чворови се визуелно представљају као тачке, а гране као линије које повезују чворове. Граф је усмерен уколико је E скуп уређених парова из V. Тада се грана из скупа E представља као линија са стрелицом од чвора који је први елемент пара до чвора који је други елемент пара. У супротном, када је E скуп неуређених парова из V, граф је неусмерен, а гране се представљају као обичне линије. У овом раду разматрају се само неусмерени графови.

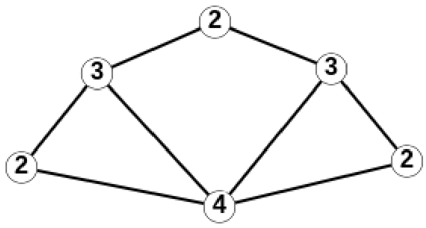
Чворови x и y неког графа су суседни уколико грана $\{x, y\}$ припада том графу. Каже се да грана $\{x, y\}$ повезује чвор x и чвор y. Грана је суседна чвору уколико повезује тај чвор са неким другим. Тако је грана $\{x, y\}$ суседна



Слика 1.1: Пример простог графа

чвору x и чвору y. Степен чвора v који припада графу $G = (V, E)$ једнак је броју грана из E које су суседне са v.

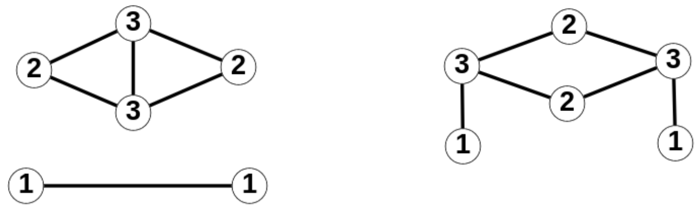
Граф је повезан ако од сваког чвора графа може да се гранама дође до било ког другог чвора графа. Изоловани чворови су чворови који нису повезани са остатком графа, односно чворови чији је степен једнак 0. Петља у графу је грана која повезује чвор са самим собом. Граф је прост уколико не садржи петље. На слици 1.1 приказан је прост неусмерен граф и на сваком чвору означен је степен тог чвора. Више о графовима може се прочитати у књизи [3].



Слика 1.2: Граф који је реализација графичког низа $\{4, 3, 3, 2, 2, 2\}$

Графички низ је низ чији елементи одговарају степенима чворова неког графа и поређани су у нерастући низ. Другим речима, низ $d = (d_1, d_2, \dots, d_n)$ је графички низ ако важи $d_1 \geq d_2 \geq \dots \geq d_n$ и постоји граф $G = (V, E)$ такав да је скуп степенова његових чворова $\{d_1, d_2, \dots, d_n\}$. У том случају каже се да је граф G реализација графичког низа $d = (d_1, d_2, \dots, d_n)$. На слици 1.2 приказан је граф који је реализација графичког низа $\{4, 3, 3, 2, 2, 2\}$.

Партиција позитивног целог броја n је низ позитивних целих бројева $(\pi_1, \pi_2, \dots, \pi_l)$ такав да важи $\pi_1 \geq \pi_2 \geq \dots \geq \pi_l$ и $\pi_1 + \pi_2 + \dots + \pi_l = n$. На пример, број 4 има 5 различитих партиција и то су: $(1, 1, 1, 1), (2, 1, 1), (2, 2), (3, 1), (4)$. Партиција $\pi = (\pi_1, \pi_2, \dots, \pi_l)$ је графичка партиција уколико је $(\pi_1, \pi_2, \dots, \pi_l)$ графички низ.



Слика 1.3: Два различита графа чији је графички низ {3, 3, 2, 2, 1, 1}

Граф $G_1 = (V_1, E_1)$ и граф $G_2 = (V_2, E_2)$ су изоморфни ако постоји функција $f : V_1 \to V_2$ која је бијекција и важи да су два чвора u и w из графа G_1 суседна ако и само ако су $f(u)$ и $f(w)$ суседни у графу G_2. За дати граф $G = (V, E)$ постоји тачно један графички низ. Међутим, за један графички низ може постојати више неизоморфних графова који му одговарају. На слици 1.3 приказана су два графа која нису изоморфна, а чији су графички низови исти.

1.2 Проблем пребројавања графичких низова

Није сваки низ графички. Пребројавање графичких низова је проблем израчунавања колико различитих графичких низова постоји за одређено n, где n представља дужину низа, односно број чворова одговарајућег графа. У овом раду разматра се пребројавање графичких низова свих простих неусмерених графова, као и пребројавање графичких низова само графова без изолованих чворова и повезаних графова.

Низ не може да буде графички уколико садржи елемент чија је вредност већа или једнака од дужине низа n, јер не постоји прост граф са степеном чвора већим од броја осталих чворова у графу. Збир свих елемената графичког низа једнак је двоструком броју грана њему одговарајућег графа и стога је увек паран. Због тога, да би низ био графички, збир његових елемената мора да буде паран број. Услов да елементи низа нису већи од $n-1$ и услов да сума елемената буде паран број нису довољни услови да низ буде графички. На пример, сума елемената низа $(3, 3, 1, 1)$ је парна и сваки његов елемент има вредност мању од 4, а тај низ није графички.

Познати су различити услови који обезбеђују да низ буде графички. Први овакав услов дефинисао је Хавел [7] (Havel, 1955.), а независно од њега и Хакими [6] (Hakimi, 1962.). Ердеш (Erdős) и Галаи (Gallai) су 1960. дали још један значајан услов да низ буде графички [5]. Сирксма (Sierksma) и Хоугевејн (Hoogeveen) [13] су доказали седам различитих услова да низ буде графички, а још један услов доказао је Неш-Вилијамс (Nash-Williams). У поглављима 2 и 3 описан је услов Хавела и Хакимија и услов Ердеша и Галаиа, као и алгоритми за проверу да ли је низ графички засновани на тим условима. У поглављу 4 описано је пребројавање графичких низова засновано на Неш-Вилијамсовом услову, док су у поглављу 5 приказане програмске реализације алгоритама и резултати њиховог упоређивања.

2. Услов Хавела и Хакимија

Хавел 1955. [7], а независно од њега и Хакими 1962. [6] објавили су услов који је потребан и довољан да низ буде графички. Овај услов је у раду [9] формулисан на следећи начин:

Теорема 1.1 *За* $n \geq 3$*, нерастући низ* $d = (d_1, d_2, \ldots, d_n)$ *је графички ако и само ако је низ* $d' = (d_2 - 1, d_3 - 1, \ldots, d_{d_1} - 1, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n)$ *графички.*

Доказ. Доказ се састоји из два дела: доказа да када је низ d' графички то повлачи да је и низ d графички и обрнутог смера.

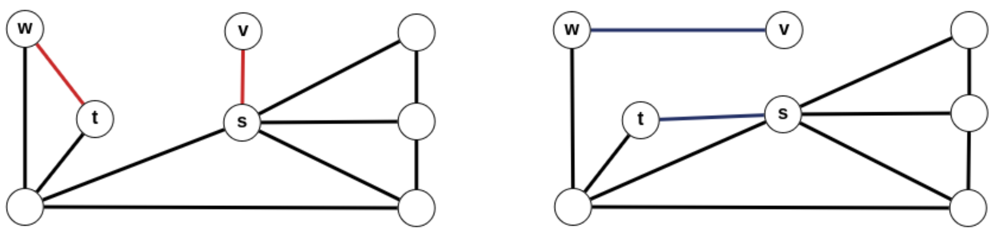
1. смер: ако је d' графички низ онда је d графички низ:
Нека је граф G' реализација графичког низа d'. Када се у граф G' дода нови чвор и повеже са d_1 чворова чији су степени $d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1$ добија се граф G чији су степени чворова $d_1, d_2, \ldots, d_n$. Граф G је реализација низа d, па је низ d графички.

2. смер: ако је d графички низ онда је d' графички низ:
Нека је граф G реализација графичког низа d и нека је његов чвор степена d_1 означен са s, чворови чији је степен $d_2, d_3, \ldots, d_{d_1}$ означени са $t_1, t_2, \ldots, t_{d_1}$ а остали чворови означени са $v_1, v_2, \ldots, v_{n-d_1}$. Разликују се два случаја.

- У првом случају, чвор s је повезан са чворовима $t_1, t_2, \ldots, t_{d_1}$. Уклањањем чвора s и његових грана добија се граф чији су степени чворова $d_2 - 1, d_3 - 1, \ldots, d_{d_1} - 1$. Такав граф је реализација низа d', па је низ d' графички.

- У другом, односно супротном случају, чвор s није повезан са неким од чворова $t_1, t_2, \ldots, t_{d_1}$. Показује се да овакав граф може да се измени тако да степени свих чворова остану исти, а чвор s буде повезан са чворовима $t_1, t_2, \ldots, t_{d_1}$ као у првом случају. Пошто је степен чвора s једнак d_1 и s није повезан са сваким од $t_1, t_2, \ldots, t_{d_1}$, онда је s повезан са неким од чворова v_i. Нека чвор t_i који није повезан са s носи ознаку t, а чвор v_i

који је повезан са s ознаку v. Пошто је степен било ког чвора v_i мањи или једнак од степена било ког чвора t_i, онда је и степен чвора v мањи или једнак од степена чвора t. Уколико је степен чвора v једнак степену чвора t, довољно је да се замене ознаке чворова v и t и добија се граф где је s повезан са свим t_i. Уколико је степен чвора v мањи од степена чвора t, онда постоји чвор w који је повезан са t, а није повезан са v. Уклањањем грана (s, v) и (t, w) и додавањем грана (s, t) и (v, w) добија се нови граф, који је такође реализација графичког низа d, а код којег је s повезан са t. Уколико чвор s није повезан са више од једног од чворова из низа $t_1, t_2, \ldots,$ t_{d_1}, онда је повезан са исто толико чворова из низа $v_1, v_2, \ldots, v_{n-d_1}$ и за сваки од неповезаних t_i чворова може да се изврши замена грана, као и у случају када је само један чвор t_i неповезан са s. Тако се добија граф код којег су сви чворови t_i повезани са чвором s, а чији је графички низ d [18]. □



Слика 2.1: Замена грана код доказа теореме Хавела и Хакимија

Пример 1. На слици 2.1 лево приказан је пример другог случаја на графу који је реализација графичког низа $(5, 4, 3, 3, 2, 2, 2, 1)$. Црвеном бојом означене су гране (s, v) и (t, w). На десној слици приказан је граф након замене грана. Плавом бојом означене су гране (s, t) и (v, w).

2.1 Алгоритам Хавела и Хакимија

Доказана теорема је основа за рекурзивни алгоритам којим се проверава да ли је низ графички. Алгоритам се може описати и кодом Алгоритам 1. У сваком кораку се из низа уклања највећи елемент d_1, после чега се d_1 највећих преосталих чланова низа смањују за 1. Уколико одузимањем настане негативан број или уколико у низу има мање елемената од вредности d_1, низ није графички. Уколико се дође до низа који садржи само нуле, низ је графички. Низ се у сваком кораку сортира како би на почетку низа били највећи елементи од којих се одузима. Дужина низа је у првом кораку једнака n, а у сваком следећем кораку је мања за по један. У сваком кораку се врши соритрање низа, те је сложеност алгоритма $O(\sum_{k=1}^{n} k \log k)$ што може да се ограничи и са $O(n^2 \log n)$.

Пример 2.

4 3 2 2 2 1
2 1 1 1 1 сортирање
2 1 1 1 1
0 0 1 1 сортирање
1 1 0 0
0 0 0 0 низ је графички

Пример 3.

3 3 1 1
2 0 0 сортирање
2 0 0
-1 -1 низ није графички

Алгоритам 1 Алгоритам Хавела и Хакимија

Улаз: низ $d = (d_1, d_2, \ldots, d_n)$
Излаз: True уколико је низ d графички, у супротном False

while True:
    if постоји $d_i < 0$ **then**:
        return False;
    сортирај низ d у нерастућем поретку;
    if $d_1 == 0$ **then**:
        return True;
    додели d_1 променљивој k и уклони d_1 из низа d;
    if $k >$ дужине низа d **then**:
        return False;
    одузми 1 од првих k елемената низа d;

Уз мале измене овај алгоритам може успут да конструише граф који одговара низу уколико је низ графички. Сваки елемент почетног низа одговара једном чвору и потребно је сваком елементу низа доделити ознаку једног чвора како би се знало на који чвор се који елемент односи након сортирања. Кад год се од елемента одузима 1, у скуп грана се додаје грана између чвора који је у том кораку уклоњен и чвора који одговара елементу од којег се одузело 1. Измењени алгоритам може да се опише и кодом Алгоритам 2.

Алгоритам 2 Алгоритам Хавела и Хакимија са конструкцијом графа

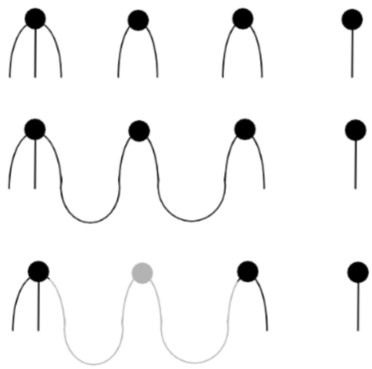
Улаз: низ $d = (d_1, d_2, \ldots, d_n)$
Излаз: $G = (V, E)$ - Граф који је реализација низа d уколико је низ d графички, False - уколико низ d није графички

трансформиши низ d у низ парова $((d_1, 1), (d_2, 2), \ldots, (d_n, n))$;
$E := \{\}$;
$V := \{1, 2, \ldots, n\}$;
while True:
    if постоји неки пар где је први елемент $d_i < 0$ **then**:
        return False;
    сортирај парове низа d опадајуће на основу првих елемената парова;
    (k, v) := први пар у низу;
    if $k == 0$ **then**:
        return (V, E);
    if $k >$ дужина низа d **then**:
        return False;
    for $i := 1$ **to** k:
        одузми 1 од првог елемента i-тог пара низа d;
        u := други елемент i-тог пара низа d;
        додај (v, u) у скуп грана G;

За елемент који се уклања из низа није неопходно изабрати највећи елемент. Теорема 1.1 важи и када се за елемент који се уклања изабере било који елемент из низа, важно је само да се од преосталих елемената бирају највећи за смањивање. Хорват (Horvát) и Моудс (Modes) у свом раду [8] формулишу теорему Хавела и Хакимија без услова да елемент d_1 буде највећи.

Теорема 1.2. *Нека је низ $d = (d_1, d_2, \ldots, d_n)$ такав да важи $d_2 \geq d_3 \geq \ldots \geq d_n$. Низ d је графички ако и само ако је низ $d' = (d_2 - 1, d_3 - 1, \ldots, d_{d_1} - 1, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n$ графички.*

Доказ Теореме 1.1 важи и за Теорему 1.2 јер се у доказу нигде не користи чињеница да је елемент d_1 највећи елемент низа. У раду [4] алгоритам Хавела и Хакимија је визуелно приказан тако што се за сваки елемент низа нацрта по један чвор из кога излази онолико грана колико одговара том елементу. Затим се у сваком кораку бира један чвор и све његове гране се спајају са гранама преосталих чворова, при чему се од преосталих чворова бирају они са највећим бројем слободних грана. Са сваким кораком смањује се број неукључених грана. Уколико после обраде последњег чвора постоје неукључене гране, низ није графички.



Слика 2.2: Алгоритам Хавела и Хакимија

Пример 4. На слици 2.2 у првом реду је приказано почетно стање за графички низ $\{3, 2, 2, 1\}$, где је за сваки елемент графичког низа нацртан по један чвор са одговарајућим бројем неповезаних грана. Затим се у другом реду обрађује један чвор (други по реду на слици 2.2) и његове гране се повезују са гранама прва два од преосталих чворова. Повезане гране и обрађени чвор се занемарују у наставку алгоритма (трећи ред слике 2.2).

У раду [9] представљен је алгоритам Хавела и Хакимија са провером парности који представља малу измену основног алгоритма Хавела и Хакимија. Измена алгоритма се састоји у томе што се прво провери да ли је сума низа парна, па тек онда уради алгоритам Хавела и Хакимија. Ова провера омогућава да се за низове чије суме нису парне већ у првом кораку види да нису графички. Као и код оригиналног алгоритма Хавела и Хакимија, сложеност је $O(n^2 \log n)$.

2.2 Измена алгоритма ради добијања повезаног графа

Алгоритам Хавела и Хакимија са конструкцијом понекад као резултат даје граф који није повезан, иако је могуће од датог графичког низа конструисати и повезани граф. Хорват и Моудс [8] дају мало измењен алгоритам Хавела и

Хакимија, који као резултат увек даје повезан граф уколико је било могуће конструисати повезан граф од датог графичког низа. Измена алгоритма је у томе што се у сваком кораку за чвор који треба да се уклони из низа бира чвор који има најмањи број преосталих грана. Да би се доказало да овако измењени алгоритам увек враћа повезан граф за графички низ који је потенцијално повезан, показује се да се овим алгоритмом из скупа чворова који имају слободне гране у сваком кораку одузима највише један чвор. Изузетак је последњи корак када низ постане (1, 1) и тада се у једном кораку два последња чвора смање до степена 0. У обрнутом смеру овај поступак повезује један по један чвор са већ постојећим графом, те не може да направи неповезан граф. Пошто се за уклањање увек бира чвор најмањег степена, чворови са којим се он повезује имају већи или једнак степен од њега и од њих се одузима 1. Дакле, остали чворови могу да буду уклоњени само када им је степен 1, а тада је и чвор који се уклања степена 1 јер је он најмањи. У том случају низ је сачињен од јединица. Изузев низа (1, 1), низови који се састоје од јединица не могу да се реализују повезаним графом. Дакле, за низове који могу да се реализују повезаним графом алгоритам уклања највише један чвор у сваком кораку. У раду [8] дат је и формални доказ тврђења да уколико постоји повезан граф који је реализација улазног низа и уколико се у сваком кораку уклања најмањи елемент низа, резултат алгоритма ће бити повезан граф.

3. Услов Ердеша и Галаиа

Ердеш и Галаи су 1960. објавили следеће тврђење:

Теорема 2. *Нерастући низ* $d = (d_1, d_2, \ldots, d_n)$ *је графички ако и само ако је сума његових елемената парна и низ задовољава услов:*

$$\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} \min\{k, d_i\} \tag{1}$$

за све k*:* $1 \leq k \leq n-1$ [5].

Доказ. Доказ се састоји из два дела: доказа да уколико је низ графички онда за $1 \leq k \leq n-1$ важи услов (1) и обрнутог смера, односно доказа да уколико је за неки низ услов (1) испуњен за $1 \leq k \leq n-1$ и сума његових елемената је парна, тај низ је графички. Други смер је компликованије доказати од првог и објављено је више различитих решења [2, 14, 4]. У наставку је описан доказ конструкцијом, који су 2010. објавили Трипатхи (Tripathi), Венугопалан (Venugopalan) и Вест (West) [15].

1. смер: ако је низ d графички онда важи услов (1) за $1 \leq k \leq n-1$:

Нека је граф G реализација графичког низа $d = (d_1, d_2, \ldots, d_n)$. Лева страна неједнакости односно услова (1) је израз $\sum_{i=1}^{k} d_i$ који представља суму првих k чворова највећих степена. Гране које та сума садржи могу да се раздвоје у два скупа. Први скуп садржи гране које на оба краја имају неки од првих k чворова. Сваки од првих k чворова може да има највише $k-1$ грана ка преосталим чворовима из групе првих k чворова, те је допринос грана првог скупа суми степенова првих k чворова највише $k(k-1)$, а број грана првог скупа највише $k(k-1)/2$. Други скуп садржи гране које на једном крају имају неки од првих k чворова, а на другом крају неки од преосталих чворова. Чвор v_i који није један од првих k чворова може имати највише $\min\{k, d_i\}$ грана које га повезују са првих k чворова, па цео скуп чворова чији је индекс већи од k може да има највише $\sum_{i=k+1}^{n} \min\{k, d_i\}$ грана ка првих k чворова. Дакле, други скуп садржи највише $\sum_{i=k+1}^{n} \min\{k, d_i\}$ грана. На левој страни неједнакости је сума степенова првих k чворова. На десној страни неједнакости је горња граница за ову суму: збир двоструке вредности максималног броја грана између првих k чворова и вредности максималног броја грана првих k чворова са преосталим чворовима. Како је сума степенова првих k чворова увек мања или једнака од наведене горње границе, неједнакост (1) важи за све индексе k, $1 \leq k \leq n-1$.

2. смер: ако важи услов за $1 \leq k \leq n-1$ и сума низа је парна онда је низ d графички:

Претпоставља се да важи услов (1) за $1 \leq k \leq n-1$ и да је сума низа парна. У граф који садржи n чворова и ниједну грану итеративно се додају гране док се не дође до графа који је реализација низа. Графови који настају овим итеративним процесом називају се подреализације. Дакле, за нерастући низ $(d_1, d_2, \ldots, d_n)$ подреализација је граф који садржи чворове $v_1, v_2, \ldots, v_n$ тако да је за свако i степен чвора v_i мањи или једнак од d_i. За низ $(d_1, d_2, \ldots, d_n)$ чија је сума парна, реализација се добија итеративним мењањем подреализације. Почетна подреализација има n чворова и нема ниједну грану.

Критични индекс r у подреализацији је највећи индекс за који важи да је $d(v_i) = d_i$ за све $1 \leq i < r$. Догод је $r \leq n$, прави се нова подреализација код које степени чворова са индексом $\leq r$ остају исти, а разлике $d_i - d(v_i)$ код чворова индекса $\geq r$ се смањују. Када r достигне $n+1$ подреализација је реализација низа $(d_1, d_2, \ldots, d_n)$ и процес је завршен.

На почетку је $r = 1$, осим ако је низ $(d_1, d_2, \ldots, d_n)$ сачињен од нула, у ком случају је процес завршен. У току формирања нове подреализације могуће је неколико различитих случајева, при чему сваки случај подразумева да ниједан од претходних није испуњен:

- *Случај 0*

  Постоји чвор v_i који није повезан са v_r такав да је $d(v_i) < d_i$. У том случају треба додати грану $v_i \leftrightarrow v_r$.

- *Случај 1*

  Постоји чвор који није повезан са v_r али његов индекс је мањи од r, односно за њега важи $d_i = d(v_i)$. Пошто је $d_i = d(v_i) \geq d_r > d(v_r)$ постоји чвор u који је повезан са v_i а није повезан са v_r. Ако је разлика d_r и $d(v_r)$ најмање 2, онда грана $\{u, v_i\}$ може да се уклони, а да се додају две нове гране: $\{u, v_r\}$ и $\{v_i, v_r\}$. Уколико је разлика мања од 2, односно једнака 1, онда постоји још један чвор, v_k, чији је степен мањи од d_i јер је разлика од сума степена такође парна јер су суме парне. Чвор v_k има индекс k већи од r. Случај 0 није испуњен, тако да постоји грана која повезује v_r и v_k. Заменом грана $\{v_r, v_k\}$ и $\{v_i, u\}$ гранама $\{v_r, v_i\}$ i $\{v_r, u\}$ повећава се степен чвора v_r, а степени чворова са мањим индексом остају исти.

- *Случај 2*

  Пошто нису испуњени претходни случајеви, v_r је повезан са свим чворовима чији је индекс мањи од r и са свим v_i код којих је $d(v_i) < d_i$. У овом случају постоји неко v_k, за $k > r$ тако да $d(v_k) \neq \min(r, d_k)$. Случај 0 не важи, па је v_r повезан са v_k. Пошто је степен чвора v_k мањи од степена чвора v_r, постоји чвор v_i који је повезан са v_r а није повезан са v_k и има индекс мањи од r. Пошто v_i има већи степен од v_r, постоји чвор u који је повезан са v_i а није повезан са v_r. Грана $\{v_i, u\}$ мења се гранама $\{v_r, u\}$ и $\{v_i, v_k\}$.

- *Случај 3*

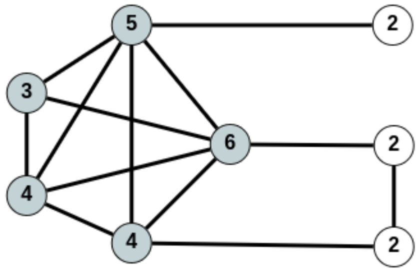
  Пошто нису испуњени претходни случајеви, v_r је повезан са свим чворовима чији је индекс мањи од r и за све v_k код којих је $k > r$ важи да је $d(v_k) = \min(r, d_k)$. Постоје чворови v_i, v_j чији су индекси мањи од r, а који нису повезани. Та два чвора су повезана са v_r јер не важи случај 1. Пошто је степен тих чворова већи од r, постоји чвор u који је повезан са v_i, а није повезан са v_r и постоји чвор w који је повезан са v_j (могуће је да је u исти чвор као и w). Чворови u и w нису повезани са v_r, а није испуњен случај 1, па су њихови индекси сигурно већи од r, односно то су чворови који још увек нису обрађени. Гране $\{u, v_i\}$ и $\{w, v_j\}$ се мењају гранама $\{u, v_r\}$ и $\{v_i, v_j\}$.

Ако нису испуњени услови ни за један од ових случајева, онда су чворови $v_1, v_2, \ldots, v_r$ сви међусобно повезани и за сваки од чворова $v_{r+1}, v_{r+2}, \ldots, v_n$ важи да је $d(v_i) = \min(r, d_i)$. По услову (1) сума степенова чворова $v_1, v_2, \ldots, v_r$ је мања или једнака од вредности $r(r-1) + \sum_{i=r+1}^{n} \min\{r, d_i\}$. Пошто је ова сума већ достигла своју горњу границу, r није критични индекс, супротно претпоставци.

$\square$

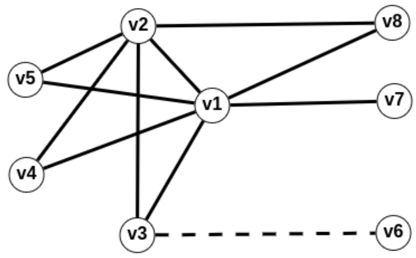
Пример 5. На слици 3.1 приказан је пример првог смера претходног доказа.

Граф на слици је реализација низа $\{6, 5, 4, 4, 3, 2, 2, 2\}$. За $k = 5$, чворови са највећим степенима су обележени сивом, а преостали чворови белом бојом. Збир степенова првих 5 чворова је 22, што представља леву страну неједнакости услова. Десна страна износи $k(k-1) + 2 + 2 + 2 = 26$, те је услов испуњен за $k = 5$.

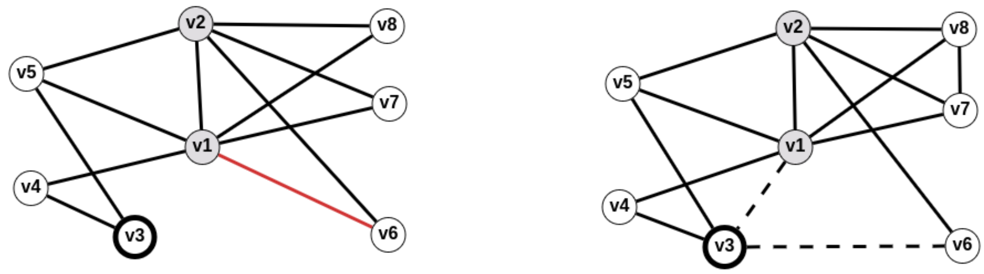


Слика 3.1: Граф са обележених првих 5 чворова највећег степена

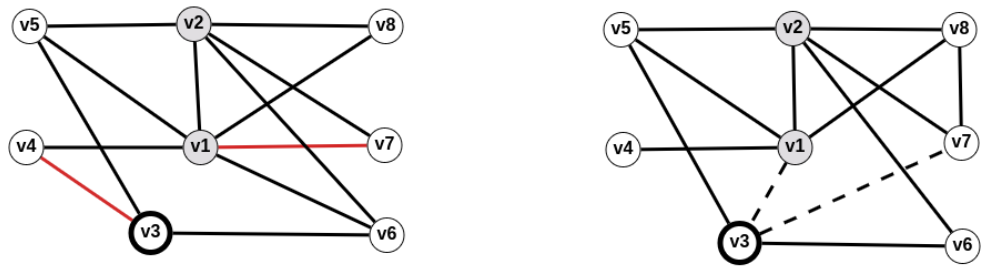
Пример 6. На слици 3.2 приказан је случај 0 из другог смера претходног доказа. За низ $\{6, 5, 4, 4, 3, 2, 2, 2\}$ приказана је подреализација након што су обрађена прва два чвора, односно када је r једнако 3. Чворови чији је индекс већи од r су v_4, v_5, v_6, v_7 и v_8. Степени ових чворова у подреализацији су редом $2, 2, 0, 1, 2$, а њима одговарајући елементи у низу су $4, 3, 2, 2, 2$, те сви чворови осим v_8 имају индекс мањи од њима одговарајућег елемента. Дакле, чвор v_3 може да се повеже са било којим од v_4, v_5, v_6, v_7 у овом кораку. На слици је испрекиданом линијом приказано његово повезивање са v_6.



Слика 3.2: Пример случаја 0



Слика 3.3: Пример случаја 1 када је $d(v_r) - d_r = 2$

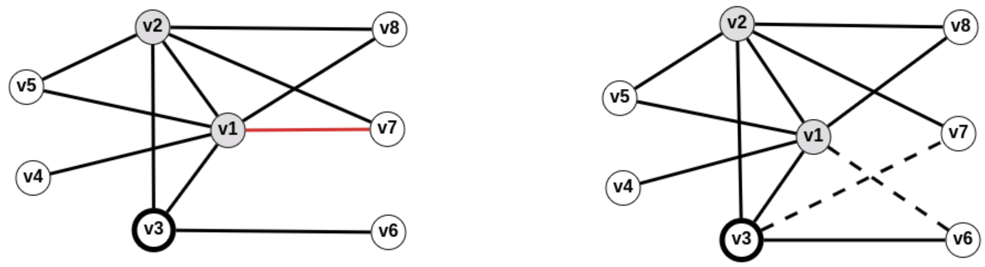


Слика 3.4: Пример случаја 1 када је $d(v_r) - d_r = 1$

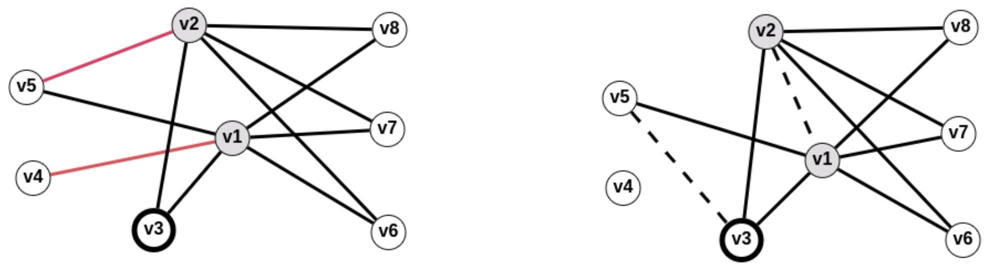
Пример 7. На сликама 3.3 и 3.4 приказан је случај 1 претходног доказа за низ $\{6, 5, 4, 4, 3, 2, 2, 2\}$ и $r = 3$. Чворови степена 6 и 5 су већ обрађени и на слици су означени сивом бојом. На слици 3.3 приказана је замена грана у случају када је $d(v_r) - d_r = 2$, а на слици 3.4 замена грана када је $d(v_r) - d_r = 1$.

Пример 8. На слици 3.5 приказан је случај 2 претходног доказа за низ $\{6, 5, 4, 2, 2, 2, 2, 1\}$ и $r = 3$. Чворови степена 6 и 5 су већ обрађени и на слици су означени сивом бојом. Чвор v_k је у овом примеру чвор v_6 који у тренутној подреализацији има степен 1, што је мање од $d(v_6) = 2$. Чвор v_i је чвор v_1 јер је повезан са v_r и није повезан са v_k, а чвор u је чвор v_7.

Пример 9. На слици 3.6 приказан је случај 3 претходног доказа на примеру низа $\{6, 5, 4, 2, 2, 2, 2, 1\}$ и $r = 3$. Чворови степена 6 и 5 су већ обрађени и на слици су означени сивом бојом.



Слика 3.5: Пример случаја 2



Слика 3.6: Пример случаја 3

3.1 Алгоритам Ердеша и Галаиа

Доказана теорема је основа за алгоритам који проверава да ли је низ графички. За све индексе низа мање од n проверава се да ли услов (1) важи. Уколико за неки индекс услов не важи, низ није графички. Алгоритам се може описати и кодом Алгоритам 3. Сложеност алгоритма је $O(n^2)$ јер постоји n корака у којима се пролази кроз цео низ дужине n како би се сабрали његови делови за проверу услова.

3.2 Скраћени алгоритам Ердеша и Галаиа

Услов (1) у теореми Ердеша и Галаиа не мора да се проверава за све индексе мање од n. Према раду Трипатхија (Tripathi) и Виџеја (Vijay) [16], довољно је проверити услов (1) за индексе $1, 2, \ldots, s$ где је s највећи индекс такав да важи $d_s \geq s - 1$.

Алгоритам 3 Алгоритам Ердеша и Галаиа

Улаз: низ $d = (d_1, d_2, \ldots, d_n)$
Излаз: True уколико је низ d графички, у супротном False

```
S := d_1 + d_2 + ... + d_n;
if S је непарно then:
    return False;
for k := 1 to n − 1:
    L := d_1 + d_2 + ... + d_k;
    D := k(k − 1) + Σ_{i=k+1}^{n} min{k, d_i};
    if L > D then:
        return False;
return True;
```

Лема 1. *Нека је низ $d = (d_1, d_2, \ldots, d_n)$ низ целих бројева такав да важи $(d_1 \geq d_2 \geq \ldots \geq d_n)$. Нека је s највећи индекс такав да важи $d_s \geq s-1$. Низ d је графички ако и само ако је сума његових елемената парна и низ задовољава услов:*

$$\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} \min\{k, d_i\}$$

за $1 \leq k \leq s$.

Доказ. За индексе k који су већи од s, важи $d_k < k-1$, па се услов (1) своди на:

$$\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} d_i$$

Уколико се разлика десне и леве стране претходне неједначине напише као функција од k добија се израз:

$$f(k) = k(k-1) + \sum_{i=k+1}^{n} d_i - \sum_{i=1}^{k} d_i$$

Када се k повећа за 1, израз постаје:

$$f(k+1) = k(k+1) + \sum_{i=k+2}^{n} d_i - \sum_{i=1}^{k+1} d_i$$

Дакле када се k повећало за 1, разлика се повећала за:

$$\begin{aligned} k(k+1) - k(k-1) + \sum_{i=k+2}^{n} d_i - \sum_{i=k+1}^{n} d_i - \sum_{i=1}^{k+1} d_i + \sum_{i=1}^{k} d_i &= 2k - 2d_{k+1} \\ &= 2(k - d_{k+1}) \end{aligned}$$

Уколико услов важи за k тј. уколико је разлика израза позитивна за k, важиће и за $k+1$ код којег је $d_{k+1} < k$, па услов (1) треба проверавати само до највећег индекса s за који важи $d_s \geq s - 1$ [16]. □

На основу овог запажања постоји скраћени алгоритам Ердеша и Галаиа, где се услов (1) проверава само до индекса s. Алгоритам се може описати и кодом Алгоритам 4. У најгорем случају индекс s може бити на самом крају низа, тако да сложеност алгоритма остаје иста као и за оригинални алгоритам Ердеша и Галаиа, односно $O(n^2)$ [9].

Алгоритам 4 Скраћени алгоритам Ердеша и Галаиа

Улаз: низ $d = (d_1, d_2, \ldots, d_n)$
Излаз: True уколико је низ d графички, у супротном False

```
S := d_1 + d_2 + ... + d_n;
if S је непарно then:
    return False;
s := n;
for i := n − 1 to 1:
    if d_i ≥ i − 1 then:
        s := i;
        break;
for k := 1 to s:
    L := d_1 + d_2 + ... + d_k;
    D := k(k − 1) + Σ_{i=k+1}^{n} min{k, d_i};
    if L > D then:
        return False;
return True;
```

3.3 Алгоритам Ердеша и Галаиа са прескакањима

Трипатхи и Вицеј су изнели још једно унапређење теореме Ердеша и Галаиа [16]. Они су показали да уколико у графичком низу постоји више истих елемената, услов из теореме Ердеша и Галаиа је потребно проверити само за јединствене елементе, при чему се за проверу бирају елементи који се налазе на позицијама са највећим индексом.

У складу са радом [16] низ који се испитује може да се опише изразом:

$$d := (d_1)_{m_1}, (d_2)_{m_2}, \ldots, (d_l)_{m_l}$$

где су d_i јединствени елементи, а свако m_i број понављања елемента d_i. Важи да је $d_1 > d_2 > \ldots > d_l$ и $m_1 + m_2 + \ldots + m_l = n$, а за свако m_i важи $m_i \geq 1$. Уводе се следеће ознаке:

$$\sigma_i := \sum_{j=1}^{i} m_j, \qquad \sigma_0 := 0 \qquad\qquad S_{r,t} := \sum_{i=r}^{t} d_i m_i$$

Теорема 3. *Низ $d := (d_1)_{m_1}, (d_2)_{m_2}, \ldots, (d_l)_{m_l}$ је графички ако и само ако је сума $S_{1,l}$ парна и низ задовољава услов (1) из теореме Ердеша и Галаиа за свако $k = \sigma_i, 1 \leq i \leq l$.*

Доказ. Потребно је доказати да уколико услов (1) важи за свако σ_i то повлачи да важи и за свако n. Трипатхи и Вицеј су ово доказали контрадикцијом. Претпостави се супротно, односно услов (1) важи за свако σ_i али не важи за неко $k = K$. Нека је K_0 најмање од свих таквих k. Индекс K_0 се може записати као збир индекса претходног јединственог елемента и броја истих елемената који следе до K_0, односно $K_0 = \sigma_i + k'$ при чему важи $1 \leq k' < m_{i+1}$ и $0 \leq i < l$. Услов не важи за K_0, те важи супротно:

$$\begin{aligned} S_{1,i} + d_{i+1}k' \quad > \quad & (\sigma_i + k')(\sigma_i + k' - 1) + (m_{i+1} - k')\min(d_{i+1}, \sigma_i + k') \\ & + m_{i+2}\min(d_{i+2}, \sigma_i + k') + \ldots + m_l \min(d_l, \sigma_i + k') \end{aligned} \tag{2}$$

Пошто је K_0 најмањи индекс за који услов не важи, за индекс $\sigma_i + k' - 1$ услов важи:

$$\begin{aligned} S_{1,i} + d_{i+1}(k' - 1) \quad \leq \quad & (\sigma_i + k' - 1)(\sigma_i + k' - 2) \\ & + (m_{i+1} - k' + 1)\min(d_{i+1}, \sigma_i + k' - 1) \\ & + m_{i+2}\min(d_{i+2}, \sigma_i + k' - 1) \\ & + \ldots + m_l \min(d_l, \sigma_i + k' - 1) \end{aligned} \tag{3}$$

Елемент d_{i+1} је већи или једнак од индекса K_0 односно $d_{i+1} \geq \sigma_i + k'$. То може да се покаже одузимањем (3) од (2):

$$\begin{aligned}
S_{1,i} + d_{i+1}k' - (S_{1,i} + d_{i+1}(k'-1)) \quad &> \\
&(\sigma_i + k')(\sigma_i + k') - (\sigma_i + k') + \\
&(m_{i+1} - k')d_{i+1} + m_{i+2}\min(d_{i+2}, \sigma_i + k') + \ldots + m_l \min(d_l, \sigma_i + k') - \\
&((\sigma_i + k')(\sigma_i + k') - 2(\sigma_i + k') - (\sigma_i + k') + 2) - \\
&((m_{i+1} - k' + 1)d_{i+1} + m_{i+2}\min(d_{i+2}, \sigma_i + k' - 1) + \ldots + m_l \min(d_l, \sigma_i + k' - 1))
\end{aligned}$$

Што се своди на:

$$d_{i+1} \quad > \quad 2(\sigma_i + k' - 1) - d_{i+1}.$$

Из претходне неједначине се даље добија $d_{i+1} > \sigma_i + k' - 1$. Дакле $d_{i+1} \geq \sigma_i + k'$ па је $\min(d_{i+1}, \sigma_i + k') = \sigma_i + k'$ и неједначина (2) се своди на:

$$\begin{aligned}
S_{1,i} + d_{i+1}k' \quad > \quad & (\sigma_i + k')(\sigma_i + k' - 1) + (m_{i+1} - k')(\sigma_i + k') + \\
& m_{i+2}\min(d_{i+2}, \sigma_i + k') + \ldots + m_l \min(d_l, \sigma_i + k')
\end{aligned}$$

Сређивањем се добија:

$$\begin{aligned}
S_{1,i} + d_{i+1}k' \quad > \quad & (\sigma_i + k')(\sigma_i + k' - 1 + m_{i+1} - k') + m_{i+2}\min(d_{i+2}, \sigma_i + k') \\
& + \ldots + m_l \min(d_l, \sigma_i + k')
\end{aligned}$$

Пошто је $\sigma_i + m_{i+1} = \sigma_{i+1}$ неједначина се даље своди на:

$$\begin{aligned}
S_{1,i} + d_{i+1}k' \quad > \quad & (\sigma_i + k')(\sigma_{i+1} - 1) + m_{i+2}\min(d_{i+2}, \sigma_i + k') \\
& + \ldots + m_l \min(d_l, \sigma_i + k')
\end{aligned} \tag{4}$$

Нека је индекс r такав да важи $d_r < \sigma_i + k' \leq d_{r-1}$. Такав индекс r постоји, јер када би $\sigma_i + k$ било мање од свих елемената низа, односно када би важило $\sigma_i + k' \leq d_{r-1}$, неједначина (4) би се свела на:

$$S_{1,i} + d_{i+1}k' \quad > \quad (\sigma_i + k')(\sigma_{i+1} - 1) + m_{i+2}(\sigma_i + k') + \ldots + m_l(\sigma_i + k')$$

Ова неједначина може даље да се среди:

$$\begin{aligned}
S_{1,i} + d_{i+1}k' \quad &> \quad (\sigma_i + k')(\sigma_{i+1} - 1 + m_{i+2} + \ldots + m_l) \\
S_{1,i} + d_{i+1}k' \quad &> \quad (\sigma_i + k')(\sigma_l - 1)
\end{aligned}$$

Ово је контрадикција, јер се зна да је $d_1(\sigma_i + k') \geq S_{1,i} + d_{i+1}k'$ и да је $d_1 \leq n - 1 = \sigma_l - 1$. Дакле постоји r за које важи $d_r < \sigma_i + k' \leq d_{r-1}$.

Пошто је $d_{i+1} \geq \sigma_i + k'$ онда је $r - 1 \geq i + 1$, односно $r \geq i + 2$. Неједначина (4) се даље своди на:

$$\begin{aligned} S_{1,i} + d_{i+1}k' \quad &> \quad (\sigma_i + k')(\sigma_{i+1} - 1) + m_{i+2}(\sigma_i + k') \\ &\quad + \ldots + m_{r-1}(\sigma_i + k') + m_r d_r + \ldots + m_l d_l \end{aligned}$$

$$\begin{aligned} S_{1,i} + d_{i+1}k' \quad &> \quad (\sigma_i + k')(\sigma_{i+1} - 1 + m_{i+2} + \ldots + m_{r-1}) + m_r d_r + \ldots + m_l d_l \\ S_{1,i} + d_{i+1}k' \quad &> \quad (\sigma_i + k')(\sigma_{r-1} - 1) + S_{r,l} \qquad (5) \end{aligned}$$

Слично неједначини (2), неједначина (3) се своди на:

$$S_{1,i} + d_{i+1}(k' - 1) \quad > \quad (\sigma_i + k' - 1)(\sigma_{r-1} - 1) + S_{r,l} \qquad (6)$$

Када се од (6) одузме (5) добија се да је $d_{i+1} \geq \sigma_{r-1}$.

С друге стране, нека је k'' највећи од свих целих бројева који су мањи или једнаки од m_{i+1} такав да услов (1) не важи за $\sigma_i + k''$. Важи $k' \leq k'' < m_{i+1}$. Пошто услов (1) не важи за $\sigma_i + k''$, важи супротно:

$$\begin{aligned} S_{1,i} + d_{i+1}k'' \quad &> \quad (\sigma_i + k'')(\sigma_i + k'' - 1) + (m_{i+1} - k'')\min(d_{i+1}, \sigma_i + k'') \\ &\quad + m_{i+2}\min(d_{i+2}, \sigma_i + k'') + \ldots + m_l \min(d_l, \sigma_i + k'') \end{aligned} \qquad (7)$$

Будући да је $\sigma_i + k''$ највећи индекс за који услов не важи, за индекс $\sigma_i + k'' + 1$ услов важи:

$$\begin{aligned} S_{1,i} + d_{i+1}(k'' + 1) \quad &\leq \quad (\sigma_i + k'')(\sigma_i + k'' + 1) \\ &\quad + (m_{i+1} - k'' - 1)\min(d_{i+1}, \sigma_i + k'' + 1) \\ &\quad + m_{i+2}\min(d_{i+2}, \sigma_i + k'' + 1) \\ &\quad + \ldots + m_l \min(d_l, \sigma_i + k'' + 1) \end{aligned} \qquad (8)$$

Важи $\sigma_i + k'' < \sigma_{i+1} \leq \sigma_{r-1} \leq d_{i+1}$. Нека је цео број s такав да је $d_s \leq \sigma_i + k'' < d_{s-1}$. Важи да је $i + 1 < s \leq r$. Једначина (7) сада може да се напише уз помоћ s:

$$\begin{aligned} S_{1,i} + d_{i+1}k'' \quad &> \quad (\sigma_i + k'')(\sigma_i + k'' - 1) + (m_{i+1} - k'')(\sigma_i + k'') \\ &\quad + m_{i+2}(\sigma_i + k'') + \ldots + m_{s-1}(\sigma_i + k'') + S_{s,l} \end{aligned} \qquad (9)$$

Као и једначина (8):

$$\begin{aligned} S_{1,i} + d_{i+1}(k'' + 1) \quad &\leq \quad (\sigma_i + k'')(\sigma_i + k'' + 1) \\ &\quad + (m_{i+1} - k'' - 1)(\sigma_i + k'' + 1) + m_{i+2}(\sigma_i + k'' + 1) \\ &\quad + \ldots + m_{s-1}(\sigma_i + k'' + 1) + S_{s,l} \end{aligned} \tag{10}$$

Када се од (10) одузме (9) добије се:

$$\begin{aligned} d_{i+1} \quad &\leq \quad 2(\sigma_i + k'') + (m_{i+1} - k'' - \sigma_i - k'' + 1) + m_{i+2} + \ldots + m_{s-1} \\ d_{i+1} \quad &\leq \quad \sigma_i + m_{i+1} + m_{i+2} + \ldots + m_{s-1} - 1 \\ d_{i+1} \quad &\leq \quad \sigma_{s-1} - 1 \end{aligned}$$

Пошто важи да је $\sigma_{s-1} - 1 < \sigma_{s-1} \leq \sigma_{r-1} \leq d_{i+1}$, претходна неједнакост је контрадикција, чиме је теорема доказана [16]. □

Доказана теорема је основа за унапређење алгоритма Ердеша и Галаиа. Уместо за свако n, услов се проверава само за $k = \sigma_i, 1 \leq i \leq l$. Унапређени алгоритам се може описати и кодом Алгоритам 5. У најгорем случају сви елементи низа су различити бројеви, па је сложеност алгоритма иста као и код оригиналног алгоритма Ердеша и Галаиа односно $O(n^2)$ [9].

3.4 Линеарни алгоритам Ердеша и Галаиа

У раду [9] описано је унапређење алгоритма Ердеша и Галаиа које има линеарну сложеност. Искоришћена је чињеница да је низ који се добија на улазу у алгоритам монотон. Наредне ознаке су у складу са радом [9]:
Нека је

$$C_k = \sum_{i=k+1}^{n} \min\{k, d_i\}$$

што је други сабирак десне стране услова (1) из теореме Ердеша и Галаиа. Унапређени алгоритам користи помоћни низ $w(d)$ који је дефинисан као $w(d) = (w_1, \ldots, w_{n-1})$, где w_k представља индекс елемента d_i који има највећи индекс од свих елемената у низу за које важи да су већи или једнаки од броја k. Уколико такав елемент не постоји у низу, онда је $w_k = 0$. Унапређени алгоритам помоћу низа $w(d)$ за константно време одређује вредности C_k помоћу $w(k)$. Тежина $w(k)$ одређује индекс елемента у низу до којег је $\min\{k, d_i\}$ једнак k, а од којег

Алгоритам 5 Алгоритам Ердеша и Галаиа са прескакањима

Улаз: низ $d = (d_1, d_2, \ldots, d_n)$
Излаз: True уколико је низ d графички, у супротном False

иницијализуј H као низ дужине $n+1$;
$H[0] := 0$;
for $i := 1$ **to** n:
    $H[i] := H[i-1] + d[i]$;
if $H[n]$ је непарно **then**:
    return False;
$k := 1$;
while $k \leq n$ и $k(k-1) < H[k]$:
    while $d_k == d_{k+1}$:
        $k := k+1$;
    $L := d_1 + d_2 + \ldots + d_k$;
    $D := k(k-1) + \sum_{i=k+1}^{n} \min\{k, d_i\}$;
    if $L > D$ **then**:
        return False;
    $k := k+1$;
return True;

је једнак d_i. H_i означава суму првих i елемената низа d.

Теорема 4. *Ако је $n \geq 1$ онда је нерастући низ $d = (d_1, d_2, \ldots, d_n)$ графички ако и само ако је збир свих елемената паран и ако за $k > w_k$ важи*

$$H_k \leq k(k-1) + H_n - H_k \tag{11}$$

а за $k \leq w_k$ важи

$$H_k \leq k(k-1) + k(w_k - 1) + H_n - H_{w_k} \tag{12}$$

Доказ. Теорема се доказује свођењем на теорему Ердеша и Галаиа. Провера парности је иста као и код теореме Ердеша и Галаиа.
Приликом провере услова (1) из теореме Ердеша и Галаиа постоје два случаја:

- Случај када је $k > w_k$. Будући да w_k представља индекс последњег елемента у низу d који је већи од k, у овом случају су сви елементи $d_{k+1}, \ldots, d_n$ мањи од k. Тада је $\min\{k, d_i\}$ за свако i једнак d_i, па је C_k једнако суми елемената низа са индексима од $k+1$ до n. C_k из услова (1) у неједнакости (11) замењено је изразом $H_n - H_k$, што је једнако збиру елемената са индексима од $k+1$ до n.

- Случај када је $k \leq w_k$. У овом случају C_k се састоји од два дела. Први део C_k су елементи $c_{k+1}, \ldots, c_{w_k}$ који су једнаки k и њих има $(w_k - k)$ па се тај део суме C_k у неједначини (12) мења изразом $k(w_k - k)$. Остатак суме чине елементи $c_{w_k}, \ldots, c_n$ за које је $d_i < w_k$, те је код њих $\min\{k, d_i\}$ једнак d_i. Овај део суме у ствари представља збир елемената низа са индексима од w_k до n, односно $H_n - H_{w_k}$.

$\square$

Алгоритам 6 Линеарни алгоритам Ердеша и Галаиа

Улаз: низ $d = (d_1, d_2, \ldots, d_n)$

Излаз: True уколико је низ d графички, у супротном False

```
иницијализуј H као низ дужине n + 1;
H[0] := 0;
for i := 1 to n:
        H[i] := H[i − 1] + d[i];
if H[n] је непарно then:
        return False;
иницијализуј w као низ дужине n;
d_0 := n − 1;
for i := 1 to n:
        if d_i < d_{i−1} then:
                for j := d_{i−1} to d_i + 1:
                        w[j] := i − 1;
                        w[d_i] := i;
for i := d_n to 1:
        w[j] := n;
for i := 1 to n:
        if i ≤ w_i then:
                if H_i ≥ i(i − 1) + i(w_i − 1) + H[n] − H[w_i] then:
                        then False;
        if i > w_i then:
                if H_i ≥ i(i − 1) + H[n] − H[i] then:
                        return False;
return True;
```

Доказана теорема је основа за унапређени алгоритам који има линеарну сложеност [9]. Прво се провери да ли је сума елемената парна. Затим се израчуна низ тежина $w(d)$. Када је низ израчунат, за свако k се на основу теореме рачуна C_k и проверава да ли важи услов. Унапређени алгоритам се може описати и кодом Алгоритам 6.

4. Алгоритми који користе графичке партиције

Фереров дијаграм (*енгл.* Ferrer diagram) партиције $(\pi_1, \pi_2, \ldots, \pi_l)$ се састоји од l редова тачака, при чему ред i садржи π_i тачака и сви редови тачака су лево поравнати. На слици 4 лево приказан је Фереров дијаграм партиције $(5, 4, 3, 3, 2, 1)$. Дарфијев квадрат (*енгл.* Durfee square) је највећи квадрат тачака који Фереров дијаграм садржи. На слици 4 десно означен је Дарфијев квадрат за партицију $(5, 4, 3, 3, 2, 1)$. Величина Дарфијевог квадрата, односно његов број редова, означава се са $d(\pi)$. Важи да у партицији π постоји $d(\pi)$ или више елемената који су већи или једнаки од $d(\pi)$. За партицију $(5, 4, 3, 3, 2, 1)$ важи $d(\pi) = 3$.

Слика 4.1: Пример Фереровог дијаграма и Дарфијевог квадрата

У Фереровом дијаграму, број тачака реда i је π_i, а број тачака колоне i означава се са π'_i. Низ $(\pi_1 - \pi'_1, \pi_2 - \pi'_2, \ldots, \pi_{d(\pi)} - \pi'_{d(\pi)})$ назива се низ узастопних рангова (*енгл.* successive ranks), а низ $(\pi'_1 - \pi_1, \pi'_2 - \pi_2, \ldots, \pi'_{d(\pi)} - \pi_{d(\pi)})$ низ узастопних негативних рангова (*енгл.* successive antiranks). Негативни ранг $\pi'_i - \pi_i$, за $1 \leq i \leq d(\pi)$ означава се и са $r_i(\pi)$, те се низ узастопних негативих рангова може записати и као $(r_1(\pi), r_2(\pi), \ldots, r_{d(\pi)}(\pi))$ [1].

Услов Неш-Вилијамса. *Партиција* $\pi = (\pi_1, \pi_2, \ldots, \pi_l)$ *неког парног целог* *броја n је графичка ако и само ако за $1 \leq k \leq d(\pi)$ важи:*

$$\sum_{i=1}^{k} r_i(\pi) \quad \geq \quad k.$$

Доказ. Неједначина $\sum_{i=1}^{k} r_i(\pi) \geq k$ може да се запише као $\sum_{i=1}^{k}(\pi_i - \pi'_i) \leq -k$. Показује се да ако услов (1) из теореме Ердеша и Галаиа важи за k, онда важи и услов $\sum_{i=1}^{k}(\pi_i - \pi_i) \leq -k$, као и да је довољно да се услов из теореме Ердеша и Галаиа провери само за индексе $1, 2, \ldots, d(\pi)$. Услов теореме Ердеша и Галаиа за партицију $(\pi_1, \pi_2, \ldots, \pi_l)$ и индекс k гласи:

$$\sum_{i=1}^{k} \pi_i \quad \leq \quad k(k-1) + \sum_{i=k+1}^{l} \min(k, \pi_i).$$

Израз $\sum_{i=k+1}^{l} \min(k, \pi_i)$ једнак је броју тачака на Фереровом дијаграму које су обухваћене квадратом чији редови имају индексе од $k+1$ до m, а колоне од 1 до k. Ова једнакост важи јер за сваки индекс i број тачака у реду i је π_i, а ограничење да се обухватају само колоне са индексима од 1 до k омогућава да се у сваком реду сабира највише k тачака. Број тачака оваквог квадрата може да се изрази тако што се тачке саберу по колонама односно као $\sum_{i=1}^{k}(\pi'_i - k)$. Дакле, услов теореме Ердеша и Галаиа за индекс k еквивалентан је услову

$$\sum_{i=1}^{k} \pi_i \quad \leq \quad k(k-1) + \sum_{i=1}^{k}(\pi'_i - k).$$

Ова неједнакост може да се сведе на услов Неш-Вилијамса:

$$\begin{aligned}
\sum_{i=1}^{k} \pi_i \quad &\leq \quad k(k-1) + \sum_{i=1}^{k} \pi'_i - \sum_{i=1}^{k} k \\
\sum_{i=1}^{k} \pi_i - \sum_{i=1}^{k} \pi'_i \quad &\leq \quad k(k-1) - k^2 \\
\sum_{i=1}^{k}(\pi_i - \pi'_i) \quad &\leq \quad -k
\end{aligned}$$

Дакле, ако је партиција $(\pi_1, \pi_2, \ldots, \pi_l)$ графичка, важи услов Неш-Вилијамса за $1 \leq k \leq d(\pi)$. Треба још да се покаже да ако услов Ердеша и Галаиа важи за $1 \leq k \leq d(\pi)$, онда важи и за $d(\pi) + 1 \leq k \leq l - 1$. За индексе k за које важи $k > d(\pi)$ и за $i \geq k + 1$ важи $\pi_i \leq \pi_{d(\pi)+1} < d(\pi) + 1 < k + 1$. Дакле, за $k > d(\pi)$ и $i \geq k + 1$ важи $\pi_i < k + 1$ и $\min(k, \pi_i) = \pi_i$. Као и код доказа скраћеног алгоритма Ердеша и Галаиа, може да се покаже да када се k повећа за 1, разлика десне и леве стране неједнакости се повећава за $2(k-\pi_{k+1})$. Пошто је $\pi_{k+1} \leq k$ и разлика се повећала, уколико неједнакост важи за $k \leq d(\pi)$ важи и за $1 \leq k \leq d(\pi)$. [12] □

У складу са радом Барнса (Barnes) и Севица (Savage) [1], нека је $P(N,k,l)$ скуп партиција броја N које имају највише l делова од којих је највећи део мањи или једнак од k. Нека је $P(N,k,l,s)$ дефинисано као празан скуп за $s < 0$, а за $s \geq 0$ као скуп:

$$\{\pi \in P(N,k,l) \quad | \quad s + \sum_{i=1}^{j} r_i(\pi) \geq j, \quad 1 \leq j \leq d(\pi)\}.$$

Нека су скуп $P'(N,k,l)$ и скуп $P'(N,k,l,s)$ подскупови скупа $P(N,k,l)$ односно скупа $P(N,k,l,s)$, који се састоје од тачно l делова и чији је највећи део тачно k. На основу дефиниције $P'(N,k,l,s)$, важи

$$P'(N,k,l,0) = \{\pi \in P'(N,k,l) \quad | \quad \sum_{i=1}^{j} r_i(\pi) \geq j, \quad 1 \leq j \leq d(\pi)\}.$$

Нека је скуп $G(N,k,l)$ скуп партиција из скупа $P(N,k,l)$ које су графичке, а скуп $G'(N,k,l)$ скуп партиција из скупа $P'(N,k,l)$ које су графичке.
Нека π припада скупу $P(N,k,l)$ и нека је α партиција које се добија од π када се обришу први ред и прва колона Фереровог дијаграма. Тада је $d(\alpha) = d(\pi) - 1$. Нека је са s означено $r_1(\pi)$. Тада према услову Неш-Вилијамса, π је графичка партиција ако и само ако је $s > 0$ и за $1 \leq j \leq d(\alpha)$ важи:

$$s + \sum_{i=1}^{j} r_i(\alpha) \quad \geq \quad j.$$

Према услову Неш-Вилијамса, партиција неког парног броја π је графичка ако и само ако за $1 \leq j \leq d(\alpha)$ важи: $\sum_{i=1}^{j} r_i(\pi) \geq j$. Дакле са $P(N,k,l,0)$ дефинисан је скуп свих графичких партиција из скупа $P(N,k,l)$, а са $P'(N,k,l,0)$ је дефинисан скуп свих графичких партиција из скупа $P'(N,k,l)$. Доказано је следеће тврђење:

Лема 2. *За парни број $N \geq 0$, важи $G'(N,k,l) = P'(N,k,l,0)$ [17] и $G(N,k,l) = P(N,k,l,0)$ [1].*

Лема 3. *За* $n > 0$ *и* $1 \leq k, l, s \leq n$ *важи:*

$$\begin{aligned} |P(n,k,l,s)| - |P'(n,k,l,s)| &= |P(n,k-1,l,s)| + |P(n,k,l-1,s)| \\ &\quad - |P(n,k-1,l-1,s)| \end{aligned}$$

Доказ. Разлика скупова $P(n,k,l,s)$ и $P'(n,k,l,s)$ састоји се од оних партиција које немају тачно l делова, односно имају $l-1$ или мање делова и оних партиција чији највећи члан нема вредност тачно k, него $k-1$ или мање.

$$P(n,k,l,s) \backslash P'(n,k,l,s) = P(n,k-1,l,s) \cup P(n,k,l-1,s)$$

Број елемената леве стране једначине једнак је $|P(n,k,l,s)| - |P'(n,k,l,s)|$. Према принципу укључења и искључења број елемената скупа који је унија скупова A и B једнак је $|A \cup B| = |A| + |B| - |A \cap B|$. Пресек скупа $P(n,k-1,l,s)$ и скупа $P(n,k,l-1,s)$ је скуп $|P(n,k-1,l-1,s)|$, тако да је по принципу укључења и искључења број елемената десне стране једначине једнак
$|P(n,k-1,l,s)| + |P(n,k,l-1,s)| - |P(n,k-1,l-1,s)|$ [1]. ☐

Наредна теорема карактерише везу између скупа P и скупа P':

Теорема 5. За $N \geq 0, 1 \leq k, l \leq N, s \geq 0$, важи

$$|P'(N,k,l,s)| = |P(N-k-l+1, k-1, l-1, s+l-k-1)|.$$

Доказ. Нека је f функција дефинисана на скупу $P'(n,k,l,s)$ тако да је за π које припада $P'(n,k,l,s)$, $f(\pi) = \alpha$, где се α добија од π када се обрише први ред и прва колона Фереровог дијаграма партиције π. Показаћемо да је $\pi \in P'(n,k,l,s)$ тачно ако и само ако $\alpha \in P(N-k-l+1, k-1, l-1, s+l-k-1)$, односно да је функција f бијекција између скупа $P'(N,k,l)$ и скупа $P(N-k-l+1, k-1, l-1)$.
Уколико је $P'(N,k,l)$ празан скуп, то је или зато што је $n < k+l-1$ или зато што је $n > kl$. Уколико је $n < k+l-1$, онда је $n-k-l+1 < 0$. Уколико је $n > kl$, онда је $n-k-l+1 > kl-k-l+1$, па је $n-k-l+1 > (k-1)(l-1)$. У оба случаја, $P(n-k-l+1, k-1, l-1)$ је празан скуп. Уколико $P'(n,k,l)$ садржи само партицију $(k,1,\ldots,1)$, онда је $f((k,1,\ldots,1)) = \lambda$. У том случају постоји један чвор степена k и $k-1$ чворова степена 1, при чему је $n = k+k-1$ и важи $l = k$. Због тога је $n-k-l+1 = 0$ и $P(n-k-l+1, k-1, l-1) = \lambda$. У осталим случајевима, $d(\alpha) = d(\pi) - 1$ и $\alpha = (\alpha_1, \ldots, \alpha_m)$, где је $m = \pi'_2 - 1$, а

$\alpha_i = \pi_{i+1} - 1$ за $1 \leq i \leq m$ и $\alpha'_i = \pi'_{i+1} - 1$ за $1 \leq i \leq d(\pi) - 1$. Осим тога, важи:

$$s + \sum_{i=1}^{j}(\pi'_i - \pi_i) =$$

$$(s + l - k) + \sum_{i=2}^{j}(\pi'_i - \pi_i) =$$

$$(s + l - k) + \sum_{i=2}^{j}((\pi'_i - 1) - (\pi_i - 1)) =$$

$$(s + l - k) + \sum_{i=1}^{j-1}(\alpha'_i - \alpha_i)$$

На основу претходног закључује се да:

$$s + \sum_{i=1}^{j}(\pi'_i - \pi_i) \geq j \iff (s + l - k - 1) + \sum_{i=1}^{j-1}(\alpha'_i - \alpha_i) \geq j - 1$$

Дакле важи

$$\pi \in P'(n, k, l, s) \iff \alpha \in P(N - k - l + 1, k - 1, l - 1, s + l - k - 1)$$

[1] □

Лема 4. $P(n, k, l) = P(n, k, l, n) = P(n, k, l, s)$ za $s \geq n$.

Доказ. За свако π које припада скупу $P(n, k, l)$ и за свако j: $i \leq j \leq d(\pi)$ важи следећа неједнакост:

$$\sum_{i=1}^{j}(\pi'_i - \pi_i - 1) \geq \sum_{i=1}^{j} -\pi_i \geq -n$$

$$n + \sum_{i=1}^{j}(\pi'_i - \pi_i) \geq 1$$

Претходна неједнакост је довољан услов да π припада $P(n, k, l, n)$, те је $P(n, k, l) \subseteq P(n, k, k, n)$. На основу дефиниције $P(n, k, l, s)$ важи да је $P(n, k, l, t) \subseteq P(n, k, l, t')$ за свако $t' \geq t$, те за $s \geq n$ важи да је $P(n, k, l, n) \subseteq P(n, k, l, s)$. На основу дефиниције скупа $P(n, k, l, s)$ важи $P(n, k, l, s) \subseteq P(n, k, l)$. Из претходног следи:

$$P(n, k, l) \subseteq P(n, k, k, n), \quad P(n, k, l, n) \subseteq P(n, k, l, s), \quad P(n, k, l, s) \subseteq P(n, k, l)$$

[1] □

Главни резултат рада [1] је наредна рекурентна формула:

Теорема 6. $P(n,k,l,s)$ може да се израчуна на следећи начин:

$|P(n,k,l,s)| =$

$$\left\{\begin{array}{ll} 0, & \text{ако је } n<0 \text{ или } k<0 \text{ или } l<0 \text{ или } s<0 \\ 1, & \text{ако је } n=0 \\ 0, & \text{ако је } k=0 \text{ или је } l=0 \\ |P(n,n,l,s)|, & \text{ако је } k>n \\ |P(n,k,n,s)|, & \text{ако је } l>n \\ |P(n,k,l,n)|, & \text{ако је } s>n \\ |P(n,k-1,l,s)|+|P(n,k,l-1,s)|-|P(n,k-1,l-1,s)| & \\ +|P(n-k-l+\ 1,k-1,l-1,s+l-k-1)| & \text{у осталим случајевима} \end{array}\right\}$$

Доказ. Први случај следи из саме дефиниције $P(n,k,l,s)$. У другом случају, величина скупа је 1 јер за $n=0$ скуп садржи само празну партицију λ. У трећем случају не постоје партиције код којих је n веће од нуле, а да је притом k или l једнак нули, те је у том случају скуп празан. У четвртом случају k је веће од n и мада у скупу највећи део може бити мањи или једнак од k, он никад не може да буде већи од n, те се k може заменити вредношћу n. На сличан начин у петом случају се уместо l може ставити n када је $l>n$, јер l никад не може да буде веће од n. У шестом случају s се може заменити вредношћу n на основу Леме 4. Последњи ред се добија када се из Леме 3 извуче $|P(n,k,l,s)|$:

$$|P(n,k,l,s)| = |P(n,k-1,l,s)|+|P(n,k,l-1,s)|-|P(n,k-1,l-1,s)|+|P'(n,k,l,s)|.$$

У претходној једнакости је на основу Теореме 5 израз $|P'(n,k,l,s)|$ замењен изразом $|P(N-k-l+1,k-1,l-1,s+l-k-1)|$. □

4.1 Алгоритам за израчунавање $|D(n)|$

Нека је $D(n)$ скуп свих графичких партиција дужине n. Приликом израчунавања $|D(n)|$ на основу $|P(n,k,l,s)|$, Ванг у свом раду [17] пре свега даје везу између скупа $D(n)$ и скупа $G(n)$. Пошто сваки елемент графичког низа дужине n има вредност између 1 и $n-1$, сваки графички низ дужине n може да се представи као графичка партиција неког парног целог броја i, са тачно n делова и највећим делом тачно k, где је $n \le i \le n(n-1)$ и $1 \le k \le n-1$. Подскуп скупа $G(n)$ који садржи партиције са тачно l делова означен је са $G'(n,l)$.

Скуп целих парних бројева који су мањи или једнаки од $n(n-1)$, а већи или једнаки од n, означен је са $I_e(n)$.

$$D(n) = \bigcup_{i \in I_e(n)} G'(i,n)$$

$$G'(i,n) = \bigcup_{k=1}^{n-1} G'(i,k,n)$$

На основу претходне две једначине следи да је:

$$D(n) = \bigcup_{i \in I_e(n)} \bigcup_{k=1}^{n-1} G'(i,k,n)$$

Пресек било која два скупа $G'(i,k,n)$ за различите вредности i и k је празан скуп, те важи да је:

$$|D(n)| = \sum_{i \in I_e(n)} |G'(i,n)|$$

$$|G'(i,n)| = \sum_{k=1}^{n-1} |G'(i,k,n)|$$

$$|D(n)| = \sum_{i \in I_e(n)} \sum_{k=1}^{n-1} |G'(i,k,n)|$$

$$|D(n)| = \sum_{i \in I_e(n)} \sum_{k=1}^{n-1} |P'(i,k,n,0)|$$

$$|D(n)| = \sum_{i \in I_e(n)} \sum_{k=1}^{n-1} |P(i-k-n+1, k-1, n-1, n-k-1)|$$

Нека је $N = n(n-1)$. Потребно је алоцирати вишедимензиони низ величине $N - n + 1 \times n - 1 \times n \times N$ и коришћењем рекурзивног алгоритма за израчунавање вредности $P(n,k,l,s)$, редом израчунати све вредности закључно са $P(N-n+1, n-1, n, N)$. После тога, на основу горњег израза, израчунава се $|D(n)|$. Алгоритам може да се опише и кодом Алгоритам 7. Најзахтевнији део алгоритма је алоцирање и попуњавање вишедимензионе табеле P. Прва и последња димензија табеле су реда $O(n^2)$, док су друга и трећа димензија реда $O(n)$, што укупно даје сложеност $O(n^6)$. Будући да је остатак алгоритма мање захтеван, $O(n^6)$ представља укупну просторну и временску сложеност целог алгоритма. Постоји могућност да се за трећу димензију алоцирају само два реда, јер су за израчунавање $|D(n)|$ потребне само вредности $P|*,*,n-1,*|$, а вредности $P|*,*,l,*|$ се рачунају помоћу вредности $P|*,*,l-1,*|$. На овај начин сложеност алгоритма може да се побољша на $O(n^5)$ [17].

Алгоритам 7 Основни Вангов алгоритам

Улаз: цео број n
Излаз: број графичких низова дужине n

1. $N := n(n-1)$;
2. алоцирај вишедимензионални низ целих бројева P величине
   $N-n+1 \times n-1 \times n \times N$;
3. попуни вишедимензионални низ P на основу Теореме 6;
4. $s := 0$;
5. **for** $i := n$ **to** N:
6.   **if** i је парно **then**:
7.   **for** $k := 1$ **to** n:
8. $s := s + P[i-k-n+1, k-1, n-1, n-k-1]$;
9. **return** s;

4.2 Унапређени алгоритам за израчунавање $|D(n)|$

Нека је са $H(n)$ означен скуп графичких низова дужине n чији је највећи елемент величине тачно $n-1$ и нека је са $L(n)$ означен скуп графичких низова дужине n чији је највећи елемент мањи од $n-1$. Скуп $D(n)$ је скуп графичких низова дужине n и код таквих низова највећи елемент може бити мањи или једнак од $n-1$, те важи $D(n) = H(n) \cup L(n)$.

Скуп графичких низова који садрже и елементе чија је вредност нула означимо са $D_0(n)$. Величина скупа $D_0(n)$ једнака је унији скупа графичких низова дужине n који не садрже ни једну нулу и скупа графичких низова дужине n који садрже најмање једну нулу. Скуп графичких низова дужине n који садрже најмање једну нулу може да се добије када се скуп $D_0(n-1)$ измени тако што се сваком низу дода нула на крај. Због тога је величина скупа графичких низова дужине n који садрже најмање једну нулу једнака $|D_0(n-1)|$, те је $|D_0(n)| = |D_0(n-1)| + |D(n)|$. На основу претходне једначине рекурзивно се долази до следеће:

$|D_0(n)| = 1$ ako je $n = 1$
$|D_0(n)| = 1 + \sum_{i=2}^{n} |D(i)|$ ako je $n \geq 2$
Наредно тврђење омогућава да се $D(n)$ израчуна на нов начин:

Лема 5. $|H(n)| = |D_0(n-1)|$.

Доказ. Из сваког низа који припада скупу $H(n)$ може да се уклони чвор степена $n-1$, чиме се добија низ који припада скупу $D(n)$. Скуп који се добија на основу скупа $H(n)$ када се сваком низу уклони највећи елемент једнак је по бројности скупу $H(n)$ и подскуп је скупа $D_0(n-1)$. С друге стране, сваки граф који је реализација низа из скупа $D_0(n-1)$ може да се измени додавањем новог чвора који се повеже са свим осталим чворовима. Такав нови граф је реализација низа из скупа $H(n)$. Скуп који се добија изменом низова из скупа $D_0(n-1)$ на овај начин, по бројности је једнак скупу $D_0(n-1)$ и представља подскуп скупа $H(n)$. Према томе, важи $|H(n)| = |D_0(n-1)|$. □

Последица. $|D(n)| = |D_0(n-1)| + |L(n)|$ за $n > 2$.

Доказ. Пошто је $|D(n)| = |H(n)| + |L(n)|$ и на основу Леме 5 важи $|H(n)| = |D_0(n-1)|$, важи $|D(n)| = |D_0(n-1)| + |L(n)|$. □

Да би се формула $|D(n)| = |H(n)| + |L(n)|$ искористила за израчунавање $|D(n)|$ потребно је изразити $|L(n)|$ на основу вредности $P(n,k,l,s)$. Нека је $L'(i,l)$ скуп графичких партиција броја i са тачно l делова и највећим делом мањим од $l-1$. Нека је $J_e(n)$ скуп свих парних целих бројева већих или једнаких од n, а мањих или једнаких од $n(n-2)$. Тада је $L(n) = \cup_{i \in J_e(n)} L'(i,n)$. С обзиром на то да су скупови с десне стране једначине раздвојени, важи да је:

$$|L(n)| = \sum_{i \in J_e(n)} |L'(i,n)|.$$

По дефиницији,

$$L'(i,n) = \bigcup_{k=1}^{n-2} G'(i,k,n)$$

$$|L'(i,n)| = \sum_{k=1}^{n-2} |G'(i,k,n)|$$

Када се ове две једначине повежу, добија се:

$$L(n) = \bigcup_{i \in J_e(n)} \bigcup_{k=1}^{n-2} G'(i,k,n)$$

$$L(n) = \bigcup_{i \in J_e(n)} \bigcup_{k=1}^{n-2} P(i-k-n+1, k-1, n-1, n-k-1).$$

Вангов алгоритам се може убрзати на основу следећег тврђења:

Лема 6. $|L'(i,n)| = |L'(n(n-1)-i,n)|$ за свако $i \in J_e(n)$.

Доказ. За сваки графички низ $(d_1, d_2, \ldots, d_n)$ који припада скупу $L'(i,n)$ важи $d_1 \geq d_2 \geq \ldots \geq d_n$, $d_1 + d_2 + \ldots + d_n = N$ и $d_1 < n-1$. Сваки такав низ може да се трансформише у низ $n-1-d_n, n-1-d_{n-1}, \ldots, n-1-d_1$. За нови низ важи $n-1-d_n \geq n-1-d_{n-1} \geq \ldots \geq n-1-d_1 > 0$. Сума новог низа је $n(n-1)-(d_1+d_2+\ldots+d_n) = n(n-1)-i$ и његов највећи елемент, $n-1-d_n$ је мањи од $n-1$, те нови низ припада скупу $L'(n(n-1)-i,n)$. Функција која трансформише низ на овакав начин је бијекција и трансформише симетрично низове са највећим вредностима у низове са најмањим вредностима елемената, и обрнуто. □

Захваљујући претходној леми могуће је израчунати величину скупа $L(n)$ на следећи начин:

За вредности n чији је остатак при дељењу са 4 једнак 2 или 3:

$$|L(n)| = 2\sum_{i \in I} \sum_{k=1}^{n-2} |P(i-k-n+1, k-1, n-1, n-k-1)|$$

За вредности n чији је остатак при дељењу са 4 једнак 0 или 1:

$$|L(n)| = 2\sum_{i \in I} \sum_{k=1}^{n-2} |P(i-k-n+1, k-1, n-1, n-k-1)| +$$

$$\sum_{k=1}^{n-2} |P(n(n-1)/2-k-n+1, k-1, n-1, n-k-1)|$$

Унапређени Вангов алгоритам може да се опише кодом Алгоритам 8. Као и код основног Ванговог алгоритма, сложеност овог алгоритма је (n^6). Прва и четврта димензија табеле P су двоструко мање у односу на основни алгоритам, те је нови алгоритам бољи од основног за константни фактор 4. Међутим, његова сложеност је и даље $\mathrm{O}(n^6)$.

Алгоритам 8 Унапређени Вангов алгоритам

Улаз: цео број n, вредности $D(1), D(2), \ldots, D(n-1)$
Излаз: број графичких низова дужине n

1. $N := n(n-1)/2$;
2. алоцирај вишедимензионални низ целих бројева P величине
   $N - n + 1 \times n - 2 \times n \times N$;
3. попуни вишедимензионални низ P на основу Теореме 6;
4. $s := 0$;
5. **for** $i := n$ **to** $N - 1$:
6.   **if** је i парно **then**:
7.   **for** $k := 1$ **to** n:
8. $s := s + P[i-k-n+1, k-1, n-1, n-k-1]$;
9. $s := 2s$;
10. **if** i је парно **then**:
11.   **for** $k := 1$ **to** n:
12. $s := s + P[N-k-n+1, k-1, n-1, n-k-1]$;
13. $D_0(n-1) := 1 + \sum_{i=2}^{n-1} |D(i)|$;
14. **return** $s + D_0(n-1)$;

4.3 Пребројавање графичких низова за повезане графове дужине n

Графички низ је потенцијално повезан ако постоји повезани граф који представља његову реализацију, а принудно повезан ако су све његове реализације повезани графови. Графички низ је потенцијално неповезан ако постоји неповезани граф који представља његову реализацију, а принудно неповезан ако су све његове реализације неповезани графови.

Нека је са $D_c(n)$ означен скуп потенцијално повезаних графичких низова и са $D_d(n)$ означен скуп принудно неповезаних графичких низова. Тада је $D(n)$ унија скупа $D_c(n)$ и скупа $D_d(n)$.

Теорема 7. Графички низ $d = (d_1, d_2, \ldots, d_n)$ из скупа $D(n)$ је потенцијално повезан ако и само ако важи:

$$\sum_{i=1}^{n} d_i \geq 2(n-1).$$

Нека је $I'_e(n)$ скуп свих парних целих бројева већих или једнаких од $2(n-1)$, а мањих или једнаких од $n(n-1)$. Скуп потенцијално повезаних графичких низова чине:

$$D_c(n) = \cup_{i \in I'_e(n)} G'(i,n) = \cup_{i \in I'_e(n)} \cup_{k=1}^{n} G'(i,k,n).$$

Скуп принудно неповезаних графичких низова чине:

$$D_d(n) = \cup_{i \in I_e(n) - I'_e(n)} G'(i,n) = \cup_{i \in I_e(n) - I'_e(n)} \cup_{k=1}^{n} G'(i,k,n).$$

Пошто ови скупови имају празан пресек, важи:

$$\begin{aligned} |D_c(n)| &= \sum_{i \in I'_e(n)} \sum_{k=1}^{n} |G'(i,k,n)| \\ &= \sum_{i \in I'_e(n)} \sum_{k=1}^{n} |P(i-k-n+1, k-1, n-1, n-k-1)| \end{aligned} \tag{13}$$

$$\begin{aligned} |D_d(n)| &= \sum_{i \in I_e(n) - I'_e(n)} \sum_{k=1}^{n} |G'(i,k,n)| \\ &= \sum_{i \in I_e(n) - I'_e(n)} \sum_{k=1}^{n} |P(i-k-n+1, k-1, n-1, n-k-1)| \end{aligned} \tag{14}$$

Величина скупа потенцијално повезаних графичких низова може да се израчуна на два начина. Уколико је већ израчунато $D(n)$, онда је $|D_c(n)|$ брже израчунати као $|D_c(n)| = |D(n)| - |D_d(n)|$ користећи формулу (14). Уколико $|D(n)|$ није израчунато, онда $|D_c(n)|$ може да се израчуна директно, преко формуле (13). У наставку су приказани псеудокодови оба начина. Величина табеле P је иста као и код претходних алгоритама, тако да је сложеност ова два алгоритма такође $\mathrm{O}(n^6)$.

Алгоритам 9 Алгоритам за израчунавање $|D_c(n)|$ преко вредности $|D(n)|$

Улаз: цео број n, вредност $D(n)$
Излаз: цео број $|D_c(n)|$

1. $N := 2(n-2)$;
2. алоцирај вишедимензионални низ целих бројева P величине
   $N - n + 1 \times n - 1 \times n \times N$;
3. попуни вишедимензионални низ P на основу Теореме 6;
4. $s := 0$;
5. **for** $i := n$ **to** $N - 1$:
6.   **if** i је парно **then**:
7.   **for** $k := 1$ **to** n:
8. $s := s + P[i-k-n+1, k-1, n-1, n-k-1]$;
9. **return** $D(n) - s$;

Алгоритам 10 Алгоритам за директно израчунавање $|D_c(n)|$

Улаз: цео број n
Излаз: цео број $|D_c(n)|$

1. $N := n(n-1)$;
2. алоцирај вишедимензионални низ целих бројева P величине
   $N - n + 1 \times n - 1 \times n \times N$;
3. попуни вишедимензионални низ P на основу Теореме 6;
4. $s := 0$;
5. **for** $i := 2(n-1)$ **to** N:
6.   **if** i је парно **then**:
7.   **for** $k := 1$ **to** n :
8. $s := s + P[i-k-n+1, k-1, n-1, n-k-1]$;
9. **return** s;

5. Програмска реализација и евалуација

У овом поглављу приказане су програмске реализације алгоритама који су обрађени у претходним поглављима и експериментално је упоређена њихова ефикасност. Алгоритми су имплементирани у програмском језику *Python*. У првом делу овог поглавља упоређују се алгоритми за израчунавање броја различитих графичких низова дужине n који могу да садрже нуле као елементе, док се у другом делу упоређују алгоритми за израчунавање броја различитих графичких низова дужине n који не садрже нуле као елементе.

5.1 Алгоритми за израчунавање $|D_0(n)|$

Алгоритми за израчунавање $|D_0(n)|$ као улазни параметар имају цео број n, а као повратну вредност цео број који је једнак величини скупа који садржи све графичке низове чија је дужина n. На основу алгоритама Хавела и Хакимија, Ердеша и Галаиа и њихових варијација, описаних у поглављима 2 и 3, направљене су функције које проверавају да ли је један низ графички. Ове функције су искоришћене за израчунавање $|D_0(n)|$ тако што се за све нерастуће низове дужине n неком од ових функција проверава да ли је низ графички и уколико јесте, резултат се увећава за један. Резултат се пре ових провера иницијализује на 0, а нерастући низови дужине n се проналазе рекурзивно.

Сложеност функција за израчунавање $|D_0(n)|$ је ограничена одозго производом сложености функције која проверава да ли је низ графички и броја различитих нерастућих низова дужине n. Постоји $\binom{2n-1}{n}$ различитих нерастућих низова дужине n, где је $\binom{x}{k} = \frac{x!}{k!(x-k)!}$. Оваква формула одговара експоненцијалној сложености, па су и функције за израчунавање $|D_0(n)|$ експоненцијалне сложености [9].

Функција *checkSequencesRecur* за улазни параметар n и функцију која јој се проследи проналази рекурзивно све неопадајуће низове дужине n и прослеђеном функцијом проверава да ли је низ графички. Параметар min задаје најмању вредност коју елемент низа може да има.

```
def checkSequencesRecur(sequence, n, k, index, Dn, minValue, func):
    if k == 0:
        if func(sequence):
            Dn[0] += 1
    if k > 0:
        if index == 0:
            maxValue = n
        else:
            maxValue = sequence[index-1] + 1
        for i in range(minValue, maxValue):
            sequence[index] = i
            checkSequencesRecur(sequence, n, k - 1, index + 1,
                                                    Dn, minValue, func)
```

На основу алгоритма Хавела и Хакимија и његових измена, имплементиране су функције *HavelHakimi* и *HavelHakimiWithParityCheck*. Одговарајуће функције које израчунавају $D_0(n)$ помоћу напред наведених функција су *Dn0HavelHakimi* и *Dn0HavelHakimiWithParityCheck*. Функција *Dn0HavelHakimi* пребројава графичке низове дужине n уз помоћ функције *HavelHakimi*. Променљива *Dn0* се иницијализује на вредност 0, али се уместо целобројне вредности користи листа дужине 1 како би се њена вредност преносила кроз рекурзивне позиве функције *checkSequencesRecur*. Затим се позива рекурзивна функција *checkSequencesRecur* тако да је минимална вредност коју елемент може да има једнака 0, а функција која се користи за проверу *HavelHakimi*. Када функција *checkSequencesRecur* заврши са радом вредност променљиве *Dn0* се враћа као повратна вредност.

```
def Dn0HavelHakimi(n):
    Dn0 = [0]
    checkSequencesRecursive([0] * n, n, n, 0, Dn0, 0, HavelHakimi)
    return Dn0[0]
```

Функција *Dn0HavelHakimiWithParityCheck* је иста као функција *Dn0HavelHakimi* с тим што се уместо прослеђивања функције *HavelHakimi* прослеђује функција *HavelHakimiWithParityCheck*. Функција *HavelHakimi* која се позива у оквиру функције *Dn0HavelHakimi* имплементира алгоритам описан у одељку 2.1. Улазни параметар функције је нерастући низ, а повратна вредност је *True* ако је улазни низ графички, а *False* ако улазни низ није графички.

```
def HavelHakimi(sequence):
    n = len(sequence)
    if n == 0:
        return True
    while True:
        sequence = sorted(sequence, reverse=True)
        if sequence[0] < 0:  return False
        elif sequence[0] == 0:  return True
        else:
            value = sequence.pop(0)
            if value > len(sequence):
                return False
            for i in range(value):
                sequence[i] -= 1
                if sequence[i] < 0:
                    return False
```

Функција *HavelHakimiWithParityCheck* је иста као функција *HavelHakimi*, изузев што је на почетку функције додата провера да ли је сума низа паран број:

```
    if sum(sequence) % 2 == 1:
        return False
```

Као што су имплементације алгоритама заснованих на теореми Хавела и Хакимија употребљене за израчунавање $|D_0(n)|$, тако су употребљене имплементације алгоритама заснованих на теореми Ердеша и Галаиа. Функција *Dn0ErdosGallai* користи функцију *ErdosGallai*, функција *Dn0ErdosGallaiShort* функцију *ErdosGallaiShort*, функција *Dn0ErdosGallaiJumping* функцију *ErdosGallaiJumping*, а функција *Dn0ErdosGallaiLinear* користи функцију *ErdosGallaiLinear*. У наставку је приказана имплементација функције *ErdosGallai* на основу алгоритма из одељка 3.1.

```
def ErdosGallai(sequence):
    n = len(sequence)
    if n == 0:
        return True
    if sum(sequence) % 2 == 1:
        return False
    for k in range(1, n+1):
        left = sum(sequence[0:k])
        Ck = [x if x < k else k for x in sequence[k:n]]
        right = k*(k-1) + sum(Ck)
        if left > right:
            return False
    return True
```

Слично претходној, функција *ErdosGallaiShort* представља имплементацију одговарајућег алгоритма датог у одељку 3.2. Следи њен приказ:

```
def ErdosGallaiShort(sequence):
    n = len(sequence)
    if sum(sequence) % 2 == 1:
        return False
    s = n - 1
    for i in range(n - 1, 0, -1):
        if sequence[i] >= i:
            s = i+1
            break
    for k in range(1, s+1):
        left = sum(sequence[0:k])
        Ck = [x if x < k else k for x in sequence[k:n]]
        right = k*(k-1) + sum(Ck)
        if left > right:
            return False
    return True
```

Алгоритам Ердеша и Галаиа из одељка 3.3 имплементиран је функцијом *ErdosGallaiJumping* чији приказ следи:

```
def ErdosGallaiJumping(sequence):
```

```
    n = len(sequence)
    if sum(sequence) % 2 == 1:
        return False
    for k in range(1, n):
        if sequence[k-1] != sequence[k]:
            left = sum(sequence[0:k])
            Ck = [x if x < k else k for x in sequence[k:n]]
            right = k*(k-1) + sum(Ck)
            if left > right:
                return False
    return True
```

Функција *ErdosGallaiLinear* представља имплементацију линеарног алгоритма Ердеша и Галаиа из одељка 3.4. У наставку је приказ функције:

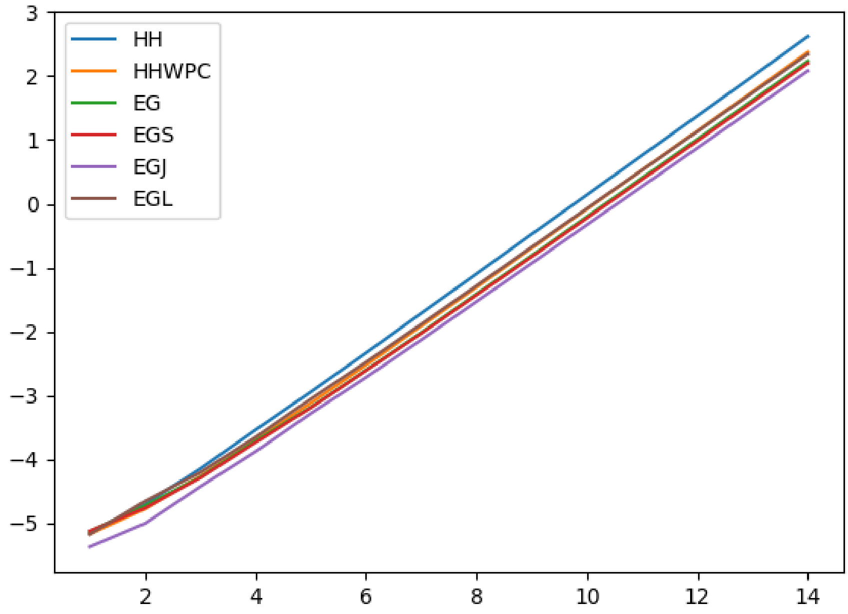
```
def ErdosGallaiLinear(sequence):
    n = len(sequence)
    H = [0]*(n+1)
    H[0] = 0
    for i in range(1, n+1):
        H[i] = (H[i-1] + sequence[i-1])
    if H[n] % 2 == 1:
        return False

    w = [0] * n
    for i in range(0, n):
        if sequence[i] < sequence[i-1]:
            j = sequence[i-1]
            while j >= sequence[i]:
                w[j] = i
                j -= 1
            w[sequence[i]] = i+1
    j = sequence[n-1]
    while j > 0:
        w[j] = n
        j -= 1
```

```
    for k in range(1, n + 1):
        left = H[k]
        if k <= w[k]:
            right = k * (k - 1) + k*(w[k] - k) + H[n] - H[w[k]]
        else:
            right = k * (k - 1) + H[n] - H[k]
        if left > right:
            return False
    return True
```

У табели 5.1 приказана су времена извршавања различитих функција у односу на цео број n који функције примају као улазни параметар. Функције које су упоређене су: *Dn0HavelHakimi*, *Dn0HavelHakimiWithParityCheck*, *Dn0ErdosGallai*, *Dn0ErdosGallaiShort*, *Dn0ErdosGallaiJumping* и *Dn0ErdosGallai-Linear*. Овим редом су и приказане у табели. Резултати израчунавања су добијени на процесору Intel Pentium брзине 1.90GHz на 64-битном Linux оперативном систему. Најбржа функција је *Dn0ErdosGallaiJumping*, док је најспорија функција *Dn0HavelHakimi*. На слици 5.1 дат је графички приказ података из табеле 5.1. На y оси дијаграма налазе се логаритми времена извршавања функција.

Табела 5.1: Дужина извршавања алгоритама за израчунавање $D_0(n)$ у секундама

n	HH	HHWPC	EG	EGS	EGJ	EGL
1	6.8e-6	6.72e-6	7.34e-6	7.47e-6	4.24e-6	6.65e-6
2	1.96e-5	1.66e-5	1.87e-5	1.71e-5	9.83e-6	2.19e-5
3	7.07e-5	5.4e-5	5.19e-5	5.09e-5	3.7e-5	6.18e-5
4	2.86e-4	2e-4	1.92e-4	1.81e-4	1.3e-4	2.2e-4
5	1.12e-3	7.57e-4	6.47e-4	6.37e-4	5.18e-4	8.75e-4
6	4.59e-3	2.95e-3	2.46e-3	2.42e-3	1.9e-3	3.29e-3
7	1.92e-2	1.21e-2	9.52e-3	9.26e-3	7.35e-3	1.29e-2
8	7.95e-2	4.89e-2	3.83e-2	3.69e-2	2.92e-2	5.18e-2
9	3.32e-1	2e-1	1.52e-1	1.46e-1	1.16e-1	2.08e-1
10	1.37	8.22e-1	6.23e-1	5.93e-1	4.66e-1	8.42e-1
11	5.71	3.37	2.52	2.37	1.86	3.35
12	2.36e+1	1.38e+1	1.01e+1	9.57e+0	7.45	1.35e+1
13	9.86e+1	5.68e+1	4.16e+1	3.88e+1	2.99e+1	5.46e+1
14	4.15e+2	2.36e+2	1.69e+2	1.57e+2	1.2e+2	2.19e+2



Слика 5.1: Логаритми времена извршавања функција у односу на улазни параметар n

5.2 Алгоритми за израчунавање $|D(n)|$

Алгоритми за израчунавање $|D(n)|$ на улазу добијају цео број n и враћају цео број који је једнак величини скупа који садржи графичке низове који не садрже елемент 0 и чија је дужина n. Као и код израчунавања $|D_0(n)|$, и при израчунавању $|D(n)|$ коришћене су функције које имплементирају алгоритме описане у поглављима 2 и 3 тј. функције: *HavelHakimi*, *HavelHakimiWithParityCheck*, *ErdosGallai*, *ErdosGallaiShort*, *ErdosGallaiJumping* и *ErdosGallaiLinear*. Одговарајући алгоритми за израчунавање $|D(n)|$ који користе ове функције су редом: *DnHavelHakimi*, *DnHavelHakimiWithParityCheck*, *DnErdosGallai*, *DnErdosGallaiShort*, *DnErdosGallaiJumping* и *DnErdosGallaiLinear*.

За разлику од функције *Dn0HavelHakimi* која позива функцију *checkSequencesRecursive* са параметром за минималну вредност елемента који низ може да садржи једнаким 0, код функција за пребројавање скупа $D(n)$ овај параметар је постављен на 1. Остатак функција се не разликује од њима одговарајућих функција из одељка 5.1. Као пример следи функција *DnHavelHakimi*.

```
def DnHavelHakimi(n):
    Dn = [0]
    checkSequencesRecursive([0] * n, n, n, 0, Dn, 1, HavelHakimi)
    return Dn[0]
```

Функција *DnWang* је имплементација алгоритма описаног у одељку 4.1. Ова функција не користи помоћне функције и не излистава све низове дужине n, већ директно рачуна вредност $D(n)$ за улазни параметар n.

```
def DnWang(n):
    N = n*(n-1)
    P = np.zeros((N-n+2, n, n+1, N+1), int)

    # popunjavanje tabele P:
    for i in range(N-n+2):
      for k in range(n):
        for l in range(n+1):
          for s in range(N+1):
            if i < 0 or k < 0 or l < 0 or s < 0:
                P[i, k, l, s] = 0
            elif i == 0:
                P[i, k, l, s] = 1
            elif k == 0 or l == 0:
                P[i, k, l, s] = 0
            elif k > i:
                P[i, k, l, s] = P[i, i, l, s]
            elif l > i:
                P[i, k, l, s] = P[i, k, i, s]
            elif s > i:
                P[i, k, l, s] = P[i, k, l, i]
            else:
                P[i, k, l, s] = P[i, k - 1, l, s] +
                    P[i, k, l - 1, s] - P[i, k - 1, l - 1, s]
                    + P[i - k - l + 1, k - 1, l - 1, s + l - k - 1]
```

```
    s = 0
    for i in range(n, N+1):
        if i % 2 == 0:
            for k in range(1, n):
                s = s + P[i - k - n + 1, k-1, n-1, n-k-1]
    return s
```

Функција *DnWangImproved* је имплементација алгоритма описаног у одељку 4.2 и представља унапређење претходне функције *DnWang*. Уз помоћ вредности $sumD0 = |D_0(1)|+|D_0(2)|+\ldots+|D_0(n-1)|$ коју прима као аргумент, функција *DnWangImproved* израчунава резултат на бржи начин. Попуњавање вишедимензионог низа P је исто као и код функције *DnWang* и стога је изостављено из наредног приказа имплементације функције *DnWangImproved*.

```
def DnWangImproved(n, sumD0):
    N = n*(n - 1)//2
    P = np.zeros((N-n+2, n, n+1, N+1), int)

    # popunjavanje tabele P

    s = 0
    for i in range(n, N):
        if i % 2 == 0:
            for j in range(1, n-1):
                    s = s + P[i - j - n + 1, j - 1, n - 1, n - j - 1]
    s = 2 * s
    if N % 2 == 0:
        for j in range(1, n-1):
            s = s + P[N - j - n + 1, j - 1, n - 1, n - j - 1]
    return s + sumD0
```

У табели 5.2 приказана су времена извршавања различитих функција у односу на цео број n који функције примају као улазни параметар, док је на слици 5.2 дат графички приказ логаритамских вредности података из табеле 5.2. Може да се закључи да је најспорији алгоритам Хавела и Хакимија. Најбржи алгоритам за $n > 12$ је унапређени Вангов алгоритам. Међутим, унапређени Вангов алгоритам користи већ израчунату вредност $|D_0(1)| + |D_0(2)| +$

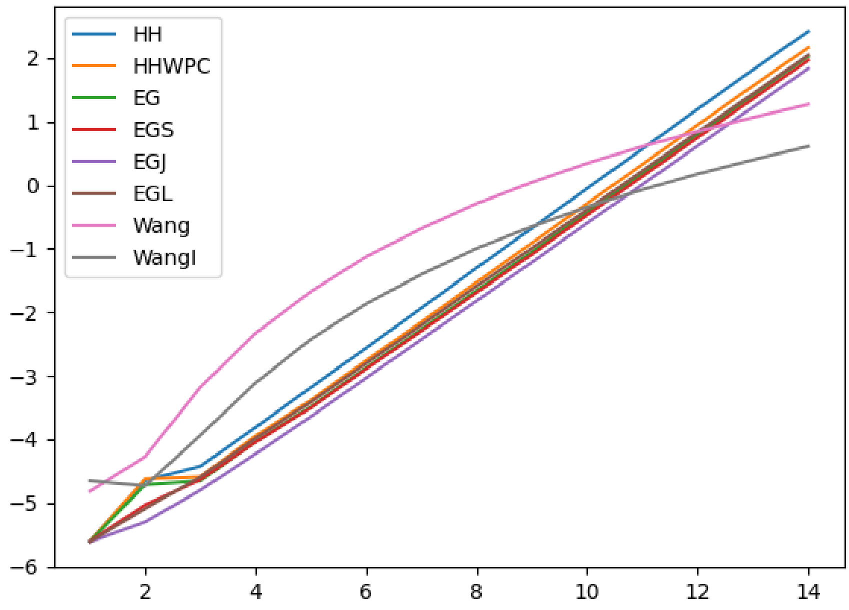
Табела 5.2: Дужина извршавања алгоритама за израчунавање $D(n)$ у секундама

n	HH	HHWPC	EG	EGS	EGJ	EGL	Wang	WImp
1	2.5e-6	2.5e-6	2.5e-6	2.5e-6	2.4e-6	2.4e-6	1.5e-5	2.2e-5
2	2.3e-5	2.4e-5	1.9e-5	9.3e-6	5e-6	8.1e-6	5.3e-5	1.8e-5
3	3.8e-5	2.6e-5	2.2e-5	2.3e-5	1.6e-5	2.6e-5	6.6e-4	1.1e-4
4	1.5e-4	1.1e-4	9.2e-5	9.1e-5	5.9e-5	1e-4	4.7e-3	7.7e-4
5	6.5e-4	4.1e-4	3.3e-4	3.1e-4	2.3e-4	3.8e-4	2.1e-2	3.7e-3
6	2.7e-3	1.7e-3	1.3e-3	1.3e-3	9.2e-4	1.6e-3	7.5e-2	1.3e-2
7	1.1e-2	7e-3	5.3e-3	5e-3	3.7e-3	6.2e-3	2.1e-1	3.9e-2
8	5e-2	3e-2	2.2e-2	2e-2	1.5e-2	2.6e-2	5.1e-1	1e-1
9	2.1e-1	1.2e-1	8.9e-2	8.1e-2	6.1e-2	1e-1	1.09	2.2e-1
10	8.8e-1	5.1e-1	3.7e-1	3.4e-1	2.5e-1	4.1e-1	2.19	4.5e-1
11	3.69	2.1	1.51	1.36	1.01	1.66	4.05	8.43e-1
12	1.6e+1	8.81	6.22	5.56	4.12	6.79	6.95	1.48
13	6.5e+1	3.6e+1	2.5e+1	2.3e+1	1.7e+1	2.7e+1	1.1e+1	2.48
14	2.6e+2	1.4e+2	1.1e+2	9.2e+1	6.8e+1	1.1e+2	1.8e+1	4.06

Табела 5.3: Вредности $D_0(n)$ и $D(n)$ за различите дужине низа n

n	1	2	3	4	5	6	7	8	9	10	11	12
$D_0(n)$	1	2	4	11	31	102	342	1213	4361	16016	59348	222117
$D(n)$	0	1	2	7	20	71	240	871	3148	11655	43332	162769

$\ldots + |D_0(n-1)|$ и када би се урачунало и време потребно да се израчунају вредности $|D_0(1)|, |D_0(2)|, \ldots, |D_0(n-1)|$, био би спорији од основног Ванговог алгоритма. Од алгоритама који израчунавају вредност $D(n)$ само на основу вредности n, најбржи је основни Вангов алгоритам, а затим алгоритам Ердеша и Галаиа са прескакањима. У табели 5.3 приказане су вредности $D_0(n)$ и $D(n)$ за различите дужине низа n за $n = 1, 2, \ldots, 12$. Ванг је захваљујући својим алгоритмима израчунао ове вредности до $n = 302$ [17]. На пример, за $n = 50$, ове вредности су $D(n) = 4521950466675832179818009402$ и $D_0(n) = 6058319744932638039744838504$.



Слика 5.2: Логаритми времена извршавања функција у односу на улазни параметар n

6. Закључак

Графички низови су нерастући низови чији елементи представљају степене неког графа. Да би низ био графички потребно је да испуњава одређене услове. Неке од најзначајних услова да низ буде графички објавили су Хавел и Хакими, Ердеш и Галаи и Неш-Вилијамс. Ови услови су основа за различите алгоритме који израчунавају број графичких низова дужине n. У овом раду истражени су алгоритми за израчунавање броја графичких низова дужине n који дозвољавају нуле као елементе низа, графичких низова дужине n који не садрже нуле као елементе низа и графичких низова дужине n за које постоји повезан граф као реализација. Најефикаснији алгоритми за сваки од ова три проблема су алгоритми из Ванговог рада [17]. Ови алгоритми користе формуле које су у свом раду Барнс и Севиџ [1] извели из Неш-Вилијамсовог услова и имају сложеност $\mathrm{O}(n^5)$. Иако је овим алгоритмима ефикасност израчунавања броја графичких низова знатно побољшана, њихова реализација је и даље превише временски захтевна, тако да је за сада израчунат број графичких низова само до вредности $n = 302$ [17].

Библиографија

[1] Tiffany M. Barnes and Carla D. Savage. A recurrence for counting graphical partitions. *the electronic journal of combinatorics*, pages R11–R11, 1995.

[2] Claude Berge. Graphs and hypergraphs. 1973.

[3] Béla Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 2013.

[4] Sheshayya A. Choudum. A simple proof of the Erdos-Gallai theorem on graph sequences. *Bulletin of the Australian Mathematical Society*, 33(1):67–70, 1986.

[5] Paul Erdős and Tibor Gallai. Gráfok előírt fokszámú pontokkal. *Matematikai Lapok*, 11:264–274, 1960.

[6] Seifollah Louis Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph. i. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506, 1962.

[7] Václav Havel. Poznámka o existenci konečných grafů. *Časopis pro pěstování matematiky*, 080(4):477–480, 1955.

[8] Szabolcs Horvát and Carl D. Modes. Connectedness matters: Construction and exact random sampling of connected networks. *Journal of Physics: Complexity*, 2(1):015008, 2021.

[9] Antal Iványi, Loránd Lucz, Tamás F. Móri, and Péter Sótér. On Erdös-Gallai and Havel-Hakimi algorithms. *Acta Universitatis Sapientiae. Informatica*, 3, 2011.

[10] Antal Iványi. Reconstruction of complete interval tournaments. ii. *Acta Universitatis Sapientiae. Mathematica*, 2(1):47–71, 2010.

[11] Hyunju Kim, Zoltán Toroczkai, Péter L. Erdős, István Miklós, and László A. Székely. Degree-based graph construction. *Journal of Physics A: Mathematical and Theoretical*, 42(39):392001, 2009.

[12] Cecil C. Rousseau and Firasath Ali. A note on graphical partitions. *Journal of Combinatorial Theory, Series B*, 64(2):314–318, 1995.

[13] Gerard Sierksma and Han Hoogeveen. Seven criteria for integer sequences being graphic. *Journal of Graph theory*, 15(2):223–231, 1991.

[14] Amitabha Tripathi and Himanshu Tyagi. A simple criterion on degree sequences of graphs. *Discrete Applied Mathematics*, 156(18):3513–3517, 2008.

[15] Amitabha Tripathi, Sushmita Venugopalan, and Douglas B. West. A short constructive proof of the Erdős–Gallai characterization of graphic lists. *Discrete mathematics*, 310(4):843–844, 2010.

[16] Amitabha Tripathi and Sujith Vijay. A note on a theorem of Erdős & Gallai. *Discrete Mathematics*, 265(1-3):417–420, 2003.

[17] Kai Wang. Efficient counting of degree sequences. *Discrete Mathematics*, 342(3):888–897, 2019.

[18] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.