

UNIVERZITA KOMENSKÉHO, BRATISLAVA  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ANALÝZA ALGORITMOV PRE  $L(2, 1)$  FARBENIE  
GRAFOV

DIPLOMOVÁ PRÁCA

2015

Michal Anderle

UNIVERZITA KOMENSKÉHO, BRATISLAVA  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ANALÝZA ALGORITMOV PRE  $L(2, 1)$  FARBENIE  
GRAFOV

DIPLOMOVÁ PRÁCA

Študijný program: Informatika  
Odbor: 2508 Informatika  
Katedra: Katedra informatiky  
Vedúci: RNDr. Michal Forišek, PhD.

Bratislava, 2015

Michal Anderle



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

---

## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Bc. Michal Anderle  
**Študijný program:** informatika (Jednoodborové štúdium, magisterský II. st., denná forma)  
**Študijný odbor:** informatika  
**Typ záverečnej práce:** diplomová  
**Jazyk záverečnej práce:** slovenský  
**Sekundárny jazyk:** anglický

**Názov:** Analýza algoritmov pre  $L(2,1)$  farbenie grafov  
*Analysis of algorithms for  $L(2,1)$  graph coloring*

**Cieľ:** V prehľadovej časti práce by autor mal odprezentovať známe techniky riešenia problému  $L(2,1)$  farbenia, s dôrazom na algoritmus od Junosza-Szaniawski et al. (TCS 2012). Prvým hlavným cieľom práce by malo byť zlepšenie dolného odhadu časovej zložitosti v najhoršom prípade pre tento algoritmus. Zvyšok práce by mal byť tvorený ďalšími vlastnými výsledkami pre vhodne zvolené problémy súvisiace s  $L(2,1)$  farbením všeobecných alebo špeciálnych grafov.

**Vedúci:** RNDr. Michal Forišek, PhD.  
**Katedra:** FMFI.KI - Katedra informatiky  
**Vedúci katedry:** doc. RNDr. Daniel Olejár, PhD.  
**Dátum zadania:** 15.12.2014

**Dátum schválenia:** 16.12.2014

prof. RNDr. Branislav Rován, PhD.  
garant študijného programu

---

študent

---

vedúci práce

# Pod'akovanie

Moje pod'akovanie patrí rodine, ktorá ma celý čas podporovala (finančne), Matfyzu, lebo je to tá najlepšia škola (aj keď z nej padajú balkóny), Trojstenu, vďaka ktorému som aký som, ale šťastný, že robím čo robím a Mišofovi, ktorý bol mojím vzorom (a školiteľom) v dobrom aj v zlom.

No a samozrejme veľká vďaka patrí mojej priateľke Katke a všetkým kamarátom, lebo vďaka nim je svet znesiteľné miesto :)

# Abstrakt

Problém  $L(2, 1)$ -farbenia je najskúmanejší problém grafového farbenia obmedzeného vzájomnou vzdialenosťou vrcholov. V roku 2012 predstavil Junosza-Szaniawski a kolektív dosiaľ najrýchlejší algoritmus riešiaci tento problém. Ako sa ukázalo, jeho časová zložitosť bola lineárna od počtu vlastných párov v danom grafe. Ostávalo však otvoreným problémom, koľko najviac vlastných párov môže mať graf s  $n$  vrcholmi, poznalo sa iba dolné ohraničenie  $\Omega(2.6117^n)$  a horné ohraničenie  $O(2.6488^n)$ .

V našej práci sa preto budeme venovať problému najväčšieho počtu vlastných párov. Zameriame sa na stromy s priemerom 6, pre ktoré dokážeme túto hodnotu asymptoticky presne –  $\Theta(6048^{n/9})$  čo je zhruba  $\Theta(2.63135^n)$ . S využitím techník prezentovaných pri stromoch s priemerom 6 následne nájdeme triedu grafov, ktorá udá lepšiu dolnú hranicu  $\Omega(2.63916^n)$ .

**Kľúčové slová:**  $L(2, 1)$ -farbenie, vlastné páry, stromy s priemerom 6

# Abstract

The  $L(2, 1)$ -labeling of a graph is one of the most researched generalizations of graph coloring; possibly the simplest version in which a constraint for vertices at distance 2 is introduced. Currently, the best known algorithm that solves this problem on general graphs is the algorithm presented by Junosza-Szaniawski et al. in 2012. The time complexity of their algorithm is linear in the number of “proper pairs” in the input graph. However, the graphs with the most proper pairs are not known yet, and therefore we do not know the exact time complexity of the algorithm by Junosza-Szaniawski et al. The best known lower bounds at the time are  $\Omega(2.6117^n)$  and  $O(2.6488^n)$ , respectively.

In this thesis we study the problem of finding graphs with as many proper pairs as possible. We prove that the worst case for trees of diameter 6 is exactly  $\Theta(6048^{n/9})$ , which is roughly  $\Theta(2.63135^n)$ . The techniques developed for trees of diameter 6 are then generalized to find an even better class of trees. The final result of the thesis is that the time complexity of the algorithm by Junosza-Szaniawski et al. is  $\Omega(2.63916^n)$ .

**Keywords:**  $L(2, 1)$ -labeling, proper pairs, trees with diameter 6

# Obsah

Úvod	1
<b>1 Prehľad problematiky <math>L(2,1)</math>-farbenia</b>	<b>3</b>
1.1 Prehľad predchádzajúcich výsledkov . . . . .	4
1.2 Algoritmus Junosza-Szaniawski et al. . . . .	5
<b>2 Vlastné páry na všeobecných stromoch</b>	<b>9</b>
2.1 Existujúce odhady počtu vlastných párov . . . . .	10
2.1.1 Horný odhad počtu vlastných párov . . . . .	10
2.1.2 Dolný odhad počtu vlastných párov . . . . .	11
2.2 Stromy s najväčším počtom vlastných párov pre malý počet vrcholov . . . . .	12
2.2.1 Generovanie všetkých neizomorfných stromov . . . . .	13
2.2.2 Počítanie vlastných párov v strome . . . . .	14
2.2.3 Implementácia . . . . .	17
2.3 Experimenty na stromoch s malým počtom vrcholov . . . . .	19
<b>3 Vlastné páry na stromoch s priemerom 6</b>	<b>25</b>
3.1 Počet vlastných párov v stromoch s priemerom 6 . . . . .	25
3.1.1 Tvar stromov s priemerom 6 . . . . .	25
3.1.2 Rátanie počtu vlastných párov . . . . .	27
3.2 Dolný odhad počtu vlastných párov . . . . .	31
3.2.1 Experimentálne hľadanie stromov . . . . .	31
3.2.2 Počet vlastných párov pre stromy s rovnakými podstromami . . . . .	32
3.2.3 Porovnanie stromov typu 7 a stromov typu 9 . . . . .	36
3.3 Nahrádzanie pravidelných podstromov . . . . .	37
3.3.1 Všeobecný počet vlastných párov . . . . .	37
3.3.2 Nahrádzanie veľkých podstromov . . . . .	39
3.3.3 Nahrádzanie malých podstromov . . . . .	41
3.4 Asymptoticky presný odhad počtu vlastných párov na stromoch priemeru 6 . . . . .	43

3.4.1	Algoritmus na nájdenie stromu priemeru 6 s najväčším počtom vlastných párov . . . . .	44
<b>4</b>	<b>Implikácie pre všeobecný prípad</b>	<b>46</b>
4.1	Limitné hľadanie hodnoty $c$ pre stromy s rovnakými podstromami . . .	46
4.2	Stromy maximalizujúce $T_1(G)$ . . . . .	49
	<b>Záver</b>	<b>52</b>



# Zoznam obrázkov

2.1	Trieda grafov slúžiaca ako dolný odhad počtu vlastných párov. . . . .	12
2.2	Stromy s najväčším počtom vlastných párov s daným počtom vrcholov.	20
2.3	Tri možné zakreslenia rekurzívneho stromu s 25 vrcholmi. V prvom prípade má podstrom veľkosti 18 a tri podstromy veľkosti 2, v druhom prípade má podstrom veľkosti 16 a štyri podstromy veľkosti 2 a v poslednom prípade má podstromy veľkostí 9, 7 a štyri podstromy veľkosti 2. .	23
3.1	Všeobecný tvar stromu s priemerom 6, ktorý má najväčší počet vlastných párov. . . . .	27
4.1	Stromy s najväčšou hodnotou $T_1(G)$ pre daný počet vrcholov. . . . .	50

# Zoznam tabuliek

2.1	Počet vlastných párov v rekurzívnych stromoch . . . . .	22
2.2	Rozloženie rekurzívnych stromov na podstromy minimalizujúce veľkosť najväčšieho podstromu . . . . .	24
3.1	Rozloženie stromov s priemerom 6 na podstromy. Dvojica v zátvorke označuje veľkosť podstromu a počet výskytov v rozložení. . . . .	32
3.2	Rozloženie stromov s priemerom 6 na podstromy. Dvojica v zátvorke označuje veľkosť podstromu a počet výskytov v rozložení. . . . .	33
3.3	Asymptotický rast počtu vlastných párov na stromoch typu $n$ . . . . .	35
3.4	Tabuľka ukazuje, koľko podstromov konkrétnej veľkosti potrebujeme, aby existovalo preusporiadanie vedúce k väčšiemu počtu vlastných párov. . . . .	43
4.1	Hodnota $T_1(G)^{1/k}$ pre $k$ vrcholový graf s najväčším $T_1(G)$ . . . . .	50

# Úvod

Napriek tomu, že problém farbenia grafov je pomerne starý a neúnavne študovaný, myšlienka farbenia obmedzeného vzájomnou vzdialenosťou vrcholov sa objavila až na konci minulého storočia. Krátko na to, ako ju v roku 1980 sformuloval Hale[4], sa pozornosť upriamila na konkrétnu inštanciu –  $L(2, 1)$ -farbenie. Hodnoty priradené vrcholom v takomto farbení, sú obmedzené nielen bezprostrednými susedmi, ale aj vrcholmi, s ktorými zdieľajú spoločného suseda. Ako sa ukázalo, takéto farbenie môže byť užitočné napríklad pri riešení *Frequency Assignment Problem*.

Problém hľadania  $L(2, 1)$ -farbenia, ktoré minimalizuje najväčšiu použitú hodnotu sa však ukázal NP-ťažký. Snahou bolo nájsť deterministické riešenie, ktorého zložitosť je síce exponenciálna, ale základ tohto exponentu je čo najmenší. Dosiaľ najlepší výsledok dosiahli Junosza-Szaniawski a kolektív, ktorí v roku 2012 uverejnili článok[7], v ktorom predstavili algoritmus na výpočet  $L(2, 1)$ -farbenia s časovou zložitosťou  $O^*(2.6488^n)$ .

Odhad zložitosti tohto algoritmu však nebol tesný. Ako sa ukázalo, časová zložitosť algoritmu závisela lineárne od počtu *vlastných párov* grafu – dvojica disjunktných podmnožín vrcholov grafu spĺňajúca dodatočné vlastnosti. No napriek tomu, že problém najväčšieho počtu vlastných párov v  $n$  vrcholovom grafe je ľahko definovateľný, jeho riešenie nie je dodnes známe. V spomínanom článku Junosza-Szaniawski a kolektív dokázali horný odhad  $O(2.6488^n)$  a dolný odhad  $\Omega(2.6117^n)$ .

Tieto dva odhady však ponúkali priestor pre zlepšenie. V našej diplomovej práci sa budeme preto zaoberať tým, koľko najviac vlastných párov môže mať  $n$  vrcholový graf.

V 1. kapitole si spravíme prehľad problematiky a zadefinujeme niekoľko kľúčových pojmov, ako sú napríklad vlastné páry. V 2. kapitole si ukážeme, že  $n$  vrcholový graf s najväčším počtom vlastných párov je strom a nájdeme takéto stromy pre niekoľko malých hodnôt  $n$ . Prihliadajúc na nájdene stromy sa v 3. kapitole pozrieme na triedu stromov s priemerom 6. Ukážeme si dolný odhad, na počet vlastných párov v takýchto stromoch, ktorý bude lepší ako doteraz existujúce výsledky a dokážeme, že tento odhad

je pre danú triedu stromov dokonca asymptoticky presný. Taktiež navrhne polynomiálny algoritmus, ktorý bude vedieť nájsť  $n$  vrcholový strom s priemerom 6 s najväčším počtom vlastných párov. V poslednej 4. kapitole potom aplikujeme postupy prezentované v 3. kapitole a odhalíme triedu stromov, ktorá určí dolné ohraňenie  $\Omega(2.63916^n)$ .

Všetky implementácie použité pri tvorení tejto diplomovej práce, ako aj niektoré vygenerované dáta, môžete nájsť na [ksp.sk/~zaba/anderle\\_diplomova\\_praca.zip](https://ksp.sk/~zaba/anderle_diplomova_praca.zip).

# Kapitola 1

## Prehľad problematiky

### $L(2,1)$ -farbenia

V tejto kapitole si zdefinujeme niekoľko kľúčových pojmov, ktoré budeme používať naprieč celým textom. Takisto sa bližšie pozrieme na motiváciu a ciele našej diplomovej práce, spravíme prehľad existujúcimi výsledkami a bližšie popíšeme algoritmus Junosza-Szaniawski a kolektívu, na ktorý naša práca nadväzuje.

**Definícia 1.0.1.**  $L(2,1)$ -*farbenie* grafu  $G$  je také priradenie nezáporných celých čísel vrcholom tohto grafu, že sú splnené nasledujúce dve podmienky:

- čísla priradené dvom susediacim vrcholom sa líšia aspoň o 2
- čísla priradené dvom vrcholom, ktoré majú spoločného suseda sú rozdielne

Ak sa pozrieme, na definíciu  $L(2,1)$ -farbenia, je prirodzené sa zaujímať o to, aké najväčšie číslo musíme použiť na ofarbenie grafu  $G$ . Túto hodnotu budeme označovať *rozpätie*  $L(2,1)$ -farbenia. Číslo  $\lambda(G)$  bude potom označovať hodnotu najmenšieho rozpätia spomedzi všetkých  $L(2,1)$ -farbení grafu  $G$ .

Dôvodov, prečo je zaujímavé hľadať  $L(2,1)$ -farbenia grafov s minimálnym rozpätím, je viacero. Najväčšiu súvislosť má však s problémom *Frequency Assignment Problem*, ktorého cieľom je priradiť jednotlivých vysielačom v sieti vysielačie frekvencie, pričom je nutné sa vyhnúť nežiadúcim interferenciám. Tie môžu vzniknúť, ak majú dva blízke vysielače podobnú vysielačiu frekvenciu, alebo majú viacerí susedia vysielača frekvenciu takú istú. Súvis s problémom  $L(2,1)$ -farbenia je očividný.

## 1.1 Prehľad predchádzajúcich výsledkov

Problém grafového farbenia je pomerne starý a podrobne skúmaný. Myšlienku grafového farbenia, ktoré je obmedzené vzájomnou vzdialenosťou vrcholov však predstavil Hale až v roku 1980[4]. O niečo neskôr sa pozornosť upriamila na konkrétnu inštanciu takýchto farbení –  $L(2, 1)$ -farbenie.

V roku 1992 Griggs a Yeh[3] predstavili vo svojej práci veľké množstvo odhadov časovej zložitosti a niekoľko zaujímavých hypotéz. Najzaujímavejšia z nich tvrdí, že  $\lambda(G) \leq \Delta(G)^2$ , kde  $\Delta(G)$  označuje maximálny stupeň grafu  $G$ . Napriek tomu, že sa jej overeniu venovalo značné množstvo času, ešte stále nie je vyriešená pre všeobecné grafy, aj keď bola overená na mnohých špeciálnych triedach. Zaujímavé je, že rovnosť tejto podmienky je známa len pre dva grafy – Hoffmann-Singletonov a Petersenov graf.

Väčšina prác sa samozrejme sústredila na to, ako vypočítať hodnotu  $\lambda(G)$  pre ľubovoľný graf. O tomto probléme však bolo dokázané, že je NP-ťažký[3, 2], čo podnietilo snahu dokázať aspoň čiastočné výsledky pre rôzne špeciálne triedy grafov. Pre stromy bol napríklad navrhnutý algoritmus, ktorý vypočíta hodnotu  $\lambda(G)$  v lineárnej časovej zložitosti[5].

Čo sa týka NP-ťažkých problémov, je pomerne častá snaha o nájdenie deterministického algoritmu, ktorý je síce exponenciálny, ale minimalizuje konštantu, ktorá je základom tohto exponentu. Inými slovami, chceme nájsť algoritmus s časovou zložitosťou  $O^*(c^n)$ , pričom  $c$  je čo najmenšie. Všimnime si, že sa používa  $O^*$ -notácia, ktorá zanedbáva polynomiálnu časť zložitosti.

Ako najúspešnejší prístup na zisťovanie hodnoty  $\lambda(G)$ , poprípade riešenie príslušnej rozhodovacej verzie, sa ukázalo dynamické programovanie. Na tomto princípe preto počas rokov vzniklo viacero zaujímavých algoritmov s postupne sa zlepšujúcou časovou zložitosťou.

Ako prvý sa objavil výsledok od Král[10], ktorý bol odvodený od riešenia iného problému a dosahoval zložitosť  $O^*(4^n)$ . Tento odhad bol neskôr zlepšený, pomocou presnejšieho odhadu počtu 2-pakovaní v grafe, na hodnotu  $O^*(3.8739^n)$ [6]. Po neskoršej modifikácii dosiahol tento algoritmus časovú zložitosť  $O^*(3.2361^n)$ , pričom sa určil aj dolný odhad zložitosti tohto algoritmu na  $\Omega^*(3.0731^n)$ [8, 9].

Hranicu  $O^*(3^n)$  sa podarilo dosiahnuť až v roku 2010 Petrom Rossmanithom. Svoj výsledok prezentoval na Dagstuhl Seminar 10441 Exact Complexity of NP-hard Problems. Vystala preto otázka, či je možné navrhnúť algoritmus, ktorého časová zložitosť bude mať exponent menší ako 3. V roku 2012 však Junosza-Szaniawski a kolektív predstavili deterministický algoritmus, ktorý dosahoval časovú zložitosť  $O^*(2.6488^n)$ [7]. V publikovanom článku tiež predstavili dolnú hranicu zložitosti a zložitosť pre claw-free grafy.

## 1.2 Algoritmus Junosza-Szaniawski et al.

V tejto časti sa bližšie pozrieme na samotný algoritmus, ktorý je dosiaľ najlepším riešením problému hľadania  $L(2, 1)$ -farbenia. V stručnosti si predstavíme hlavné myšlienky tohto algoritmu potrebné pre stanovenie cieľa našej diplomovej práce. Detailný popis algoritmu, ako aj formálny dôkaz správnosti sa dá nájsť v pôvodnom článku[7].

Najskôr si zdefinujeme binárny operátor  $\oplus$ , ktorý budeme využívať pri návrhu algoritmu. Tento operátor je binárna funkcia  $\oplus : \{0, \bar{0}, 1, \bar{1}\} \times \{0, 1\} \rightarrow \{0, 1, \bar{1}\}$  definovaná nasledovnou tabuľkou:

$\oplus$	0	$\bar{0}$	1	$\bar{1}$
0	0	0	1	1
1	$\bar{1}$	-	-	-

Všimnime si, že ak je druhá hodnota 1, je operátor definovaný iba v jedinom prípade.

Operátor  $\oplus$  môžeme prirodzene rozšíriť aj na vektory a množiny:

$$a_1 a_2 \dots a_n \oplus b_1 b_2 \dots b_n = \begin{cases} (a_1 \oplus b_1) \dots (a_n \oplus b_n) & \text{ak } a_i \oplus b_i \text{ je definované pre všetky } i \\ \text{nedefinované} & \text{v opačnom prípade} \end{cases}$$

$$A \oplus B = \{a \oplus b \mid a \in A, b \in B, a \oplus b \text{ je definované}\}$$

Na vstupe dostane náš algoritmus graf  $G$ , ktorý má  $n$  vrcholov označených  $v_1, v_2 \dots v_n$ . Postupne budeme počítat čiastočné  $L(2,1)$ -farbenia, pričom budeme dovoľovať čoraz väčšie a väčšie rozpätie týchto farbení. No a na počítanie čiastočných farbení s  $k$  farbami budeme využívať čiastočné farbenia s menším počtom farieb. Množinu čiastočných farbení s rozpätím  $k$  si označíme  $T_k$ . Je zjavné, že najväčšie rozpätie, ktoré môže mať  $n$  vrcholový graf je  $2n$ . To znamená, že potrebujeme vypočítat množiny  $T_0$  až  $T_{2n}$ .

Tieto množiny budú obsahovať čiastočné zafarbenia, ktoré budú reprezentované  $n$  prvkovými vektormi nad množinou  $\{0, \bar{0}, 1, \bar{1}\}$ . Presnejšie, množina  $T_k$  obsahuje vektor  $a$  práve vtedy, keď existuje čiastočné zafarbenie  $\varphi : V(G) \rightarrow \{0, 1, \bar{0}, \bar{1}\}$  spĺňajúce nasledujúce podmienky:

- $a_i = 0$  práve vtedy, keď  $\varphi$  nepriradzuje farbu vrcholu  $v_i$  a pre žiadneho suseda  $u$  vrchola  $v_i$  neplatí, že  $\varphi(u) = k$
- $a_i = \bar{0}$  práve vtedy, keď  $\varphi$  nepriradzuje farbu vrcholu  $v_i$  a existuje sused  $u$  vrchola  $v_i$ , pre ktorý platí, že  $\varphi(u) = k$
- $a_i = 1$  práve vtedy, keď  $\varphi(v_i) < k$
- $a_i = \bar{1}$  práve vtedy, keď  $\varphi(v_i) = k$

Uvedomme si, že ak sa nám podarí vypočítat všetky požadované množiny  $T_i$ , stačí nám nájsť najmenšie  $k$  také, že množina  $T_k$  obsahuje vektor nad množinou  $\{1, \bar{1}\}$ . Farbenie zodpovedajúce takému vektoru totiž priradzuje číslo ku každému vrcholu a má najmenšie rozpätie.

**Definícia 1.2.1.** *2-pakovanie grafu  $G$  nazývame takú podmnožinu vrcholov  $S$ , že vzájomná vzdialenosť ľubovoľných dvoch vrcholov v  $S$  je aspoň 3.*

Dôvod, prečo sme si zadefinovali pojem 2-pakovania je, že náš algoritmus bude pracovať s množinou všetkých 2-pakovaní grafu  $G$ , ktorú si označíme  $P$ . V tejto množine budú  $n$  prvkové vektory z množiny  $\{0, 1\}^n$ , pričom 1 na  $i$ -tom mieste tohto vektoru označuje, že vrchol  $v_i$  leží v 2-pakovaní. Algoritmus Junosza-Szaniawski a kolektívu bude počítat množinu  $T_{k+1}$  práve pomocou množín  $T_k$ ,  $P$  a operácie  $\oplus$ .

Nech  $a \in T_k$  je vektor reprezentujúci čiastočné farbenie  $\varphi$ . Nech  $p \in P$  reprezentuje 2-pakovanie  $S$  grafu  $G$ . Naším cieľom bude rozšíriť čiastočné farbenie  $\varphi$  o novú farbu  $k + 1$ . Práve preto dáme všetkým vrcholom z množiny  $S$  farbu  $k + 1$ . Vďaka tomu, že množina  $S$  je 2-pakovanie, nemajú dva vrcholy s farbou  $k + 1$  spoločného suseda,



čo spĺňa požiadavku  $L(2,1)$ -farbenia. Treba však vyriešiť situáciu, kedy by mali dva susedné vrcholy farbu líšiacu sa iba o 1.

Na to nám slúži operácia  $\oplus$ . Vektor  $a \oplus p$  bude totiž definovaný práve vtedy, ak je výsledný vektor korektné čiastočné farbenie. Ak je na  $i$ -tom mieste vektora  $p$  hodnota 1, na  $i$ -tom mieste vektora  $a$  musí byť 0. Vidíme, že ak by na  $i$ -tej pozícii vektora  $a$  bola 1 alebo  $\bar{1}$ , snažili by sme sa ofarbiť už ofarbený vrchol. A ak by tam bola hodnota  $\bar{0}$ , dali by sme farbu  $k+1$  vrcholu, ktorý susedí s vrcholom s farbou  $k$ . Takisto si všimnime, že operácia  $\oplus$  zmení všetky  $\bar{1}$  na 1, keďže tieto vrcholy už nemajú priradenú maximálnu farbu.

Jediným problémom sú hodnoty 0 a  $\bar{0}$  vo výslednom vektore, lebo tie nemusia byť určené správne. Túto vlastnosť budeme musieť ešte dodatočne skontrolovať a upraviť. Zanedbajúc tento detail je však najdôležitejšie vedieť rýchlo vypočítať množinu  $T_k \oplus P$ .

Na výpočet tohto problému použili autori článku podobný prístup, ako použil Strassen pri násobení matíc[12]. Majme ľubovoľnú množinu vektorov  $M$ . Potom si môžeme definovať nasledovnú množinu  $M_w = \{v \mid wv \in M\}$ . Ak chceme vypočítať výsledok  $A \oplus B$ , pre množiny  $A \subseteq \{0, \bar{0}, 1, \bar{1}\}^n$  a  $B \subseteq \{0, 1\}^n$ , môžeme použiť nasledovný vzorec:

$$\begin{aligned} A \oplus B &= \bigcup_{\substack{u \in \{0, \bar{0}, 1, \bar{1}\}^{k'} \\ v \in \{0, 1\}^{k'} \\ u \oplus v \text{ je definované}}} (u \oplus v)(A_u \oplus B_v) \\ &= \bigcup_{\substack{v \in \{0, 1\}^{k'} \\ w \in \{0, 1, \bar{1}\}^{k'}}} w \left[ \left( \bigcup_{\substack{u \in \{0, \bar{0}, 1, \bar{1}\}^{k'} \\ u \oplus v = w}} A_u \right) \oplus B_v \right] \end{aligned} \quad (1.1)$$

kde  $k'$  je vhodne zvolená konštanta.

To čo nás zaujíma je, akú zložitosť má výpočet množiny  $A \oplus B$ . Hlavne to, koľko rekurzívnych volaní na vypočítanie  $\oplus$  pre množiny vektorov dlhých  $n - k'$  budeme musieť urobiť. Zaujíma nás preto počet takých dvojíc  $v, w$ , že existuje aspoň jedno  $u$  spĺňajúce  $u \oplus v = w$ . Ak zafixujeme hodnotu  $v$ , tak hodnota  $v_i = 1$  implikuje  $w_i = \bar{1}$ . To znamená, že pre konkrétne  $v$  máme  $2^{k' - \|v\|}$  možných vektorov  $w$ , kde  $\|v\|$  označuje počet 1 vo vektore  $v$ .

**Definícia 1.2.2.** *Vlastný pár grafu  $G$  je dvojica disjunktných podmnožín vrcholov  $(S, X)$ , pričom množina  $S$  tvorí 2-pakovanie grafu  $G$ .*

Počet vlastných párov grafu  $G$  označujeme  $pp(G)$ .

Platí, že  $pp(n) = \max\{pp(G) \mid G \text{ má } n \text{ vrcholov}\}$

Uvedomme si, že keď počítame  $T_k \oplus P$ , tak vektor  $v$  je 2-pakovanie. To znamená, že počet rekurzívnych volaní je najviac  $pp(k')$ . Taktiež, veľkosť množiny  $T_k$  je najviac  $pp(n)$ , pretože vrcholy s hodnotou  $\bar{1}$  určujú 2-pakovanie a zvyšné vrcholy majú dve možnosti – 1 alebo  $0/\bar{0}$ .

Časovú zložitosť prezentovaného algoritmu teda môžeme vypočítať ako

$$T(n) = O^*(pp(n) + pp(k')T(n - k')) \quad (1.2)$$

, čoho úpravou dostaneme zložitosť  $O^*(pp(n(1 + \frac{1}{k'})))$ . Dosadením dostatočne veľkého  $k'$  a odhadom počtu vlastných párov, ktorý si prezentujeme v ďalšej kapitole dostaneme časovú zložitosť  $O^*(2.6488^n)$ .

# Kapitola 2

## Vlastné páry na všeobecných stromoch

V predchádzajúcej časti sme si ukázali algoritmus Junosza-Szaniawski a kolektívu na hľadanie  $L(2, 1)$ -farbenia. Jeho časová zložitosť bola priamo úmerná počtu vlastných párov v grafe, na ktorom problém riešime. Problém ale je, že hodnota  $pp(n)$  nemá uzavretý tvar a ani presný asymptotický odhad. Dokázaný dolný a horný odhad sú pritom od seba pomerne vzdialené.

V tejto kapitole sa preto pozrieme na počet vlastných párov vo všeobecných grafoch. Najskôr si prezentujeme už existujúci horný aj dolný odhad, pozrieme sa na grafy s najväčším počtom vlastných párov s malým počtom vrcholov a vyvodíme z nich nejaké všeobecné závery. Najskôr si však dokážeme dve lemy.

**Lemma 2.0.1.** *Počet vlastných párov grafu  $G$  je*

$$\sum_{S \subseteq V(G), S \text{ je 2-pakovanie}} 2^{n-|S|} \quad (2.1)$$

*Dôkaz.* Zoberme si nejaké 2-pakovanie  $S$ . Chceme preň zistiť počet vlastných párov  $(S, X)$ . Množina  $X$  môže, ale nemusí obsahovať každý zo zvyšných  $n - |S|$  vrcholov. Preto máme  $2^{n-|S|}$  možných vlastných párov s 2-pakovaním  $S$ . Ak tieto hodnoty sčítame cez všetky možné  $S$ , dostávame počet všetkých vlastných párov grafu  $G$ .  $\square$

**Lemma 2.0.2.** *Nech  $pp(n)$  je najväčší počet vlastných párov na súvislých grafoch s  $n$  vrcholmi. Potom existuje strom s  $n$  vrcholmi, ktorý má  $pp(n)$  vlastných párov.*

*Dôkaz.* Pre spor predpokladajme, že  $G$  je graf s  $pp(n)$  vlastnými párami, ktorý má spomedzi takýchto grafov najmenší počet hrán a predpokladajme, že  $G$  nie je strom. Vieme, že

$$pp(n) = pp(G) = \sum_{S \subseteq V(G), S \text{ je 2-pakovanie}} 2^{n-|S|} \quad (2.2)$$

teda počet vlastných párov priamo závisí od počtu a veľkosti 2-pakovaní. Vyberme si teraz ľubovoľnú hranu  $e$ , ktorá nie je mostom grafu  $G$ . Pre každé 2-pakovanie  $S$  v  $G$  však platí, že bude 2-pakovaním aj v  $G - \{e\}$ . Odstránenie hrany totiž neskráti vzdialenosť medzi žiadnymi dvoma vrcholmi a všetky vrcholy množiny  $S$  zostanú naďalej od seba vzdialené aspoň na vzdialenosť 3. To znamená, že graf  $G - \{e\}$  má tiež  $pp(n)$  vlastných párov, ale má menej hrán, čo je spor s výberom grafu  $G$ .  $\square$

Z lemy 2.0.2 preto vyplýva, že pri hľadaní grafov s najväčším počtom vlastných párov nám stačí uvažovať stromy, čo budeme využívať počas celej diplomovej práce.

## 2.1 Existujúce odhady počtu vlastných párov

V článku Junosza-Szaniawski a kolektívu[7] bol prezentovaný dolný aj horný odhad hodnoty  $pp(n)$ . V stručnosti si zhrnieme oba tieto odhady, aby sme ich mohli porovnávať s našimi dosiahnutými výsledkami.

### 2.1.1 Horný odhad počtu vlastných párov

**Lemma 2.1.1.** *Nech  $G$  je strom a vrcholy  $v$  a  $u$  sú dva listy tohto stromu, ktoré majú spoločného suseda  $w$ . Potom strom  $G'$ , ktorý vznikne z  $G$  odstránením hrany medzi vrcholmi  $u$  a  $w$  a pridaním hrany medzi vrcholmi  $v$  a  $u$  má aspoň toľko vlastných párov ako graf  $G$ .*

*Dôkaz.* Ukážeme, že všetky 2-pakovania grafu  $G$  sú 2-pakovania aj v grafe  $G'$ . Z tvrdenia lemy 2.0.1 potom vieme, že  $G'$  má aspoň toľko vlastných párov ako  $G$ .

Nech  $S$  je 2-pakovanie grafu  $G$ . Potom sa v množine  $S$  vyskytuje najviac jeden z vrcholov  $u$ ,  $v$  a  $w$ . Potrebujeme preto rozobrať nasledujúce tri prípady:

- $w \in S$  – nech  $x$  je ľubovoľný vrchol z  $S$  iný ako  $w$ . Vzdialenosť vrcholov  $w$  a  $x$  je v grafe  $G'$  rovnaká ako vzdialenosť v grafe  $G$ . A keďže  $S$  je v  $G$  2-pakovanie, je táto vzdialenosť väčšia ako 2. Množina  $S$  je preto 2-pakovaním aj v grafe  $G'$ .
- $v \in S$  – nech  $x$  je ľubovoľný vrchol z  $S$  iný ako  $v$ . Vzdialenosť vrcholov  $v$  a  $x$  je v grafe  $G'$  rovnaká ako vzdialenosť v grafe  $G$ . A keďže  $S$  je v  $G$  2-pakovanie, je táto vzdialenosť väčšia ako 2. Množina  $S$  je preto 2-pakovaním aj v grafe  $G'$ .

- $u \in S$  – nech  $x$  je ľubovoľný vrchol z  $S$  iný ako  $u$ . Vzdialenosť vrcholov  $u$  a  $x$  sa v grafe  $G'$  oproti vzdialenosti v grafe  $G$  zväčšila o 1. A keďže  $S$  je v  $G$  2-pakovanie, je táto vzdialenosť väčšia ako 3. Množina  $S$  je preto 2-pakovaním aj v grafe  $G'$ .

□

**Dôsledok 2.1.1.1.** *Pre každé  $n > 3$  existuje strom  $G$ , ktorý má najväčší počet vlastných párov medzi  $n$  vrcholovými stromami a žiadne dva listy nemajú spoločného suseda.*

*Dôkaz.* Každý strom s najväčším počtom vlastných párov, ktorý má aspoň 4 vrcholy vieme pomocou lemy 2.1.1 upraviť na nami požadovaný tvar bez toho, aby sme zmenšili počet vlastných párov. □

Pri dokazovaní horného ohraničenia počtu vlastných párov, budeme preto pracovať iba so stromami, v ktorých žiaden vrchol nie je susedný s dvoma alebo viacerými listami. Majme teda takýto strom  $G$ , ktorý má aspoň 3 vrcholy a označme si jeho najdlhšiu cestu  $P$ . Nech vrchol  $u$  je list cesty  $P$ , vrchol  $v$  je sused vrchola  $u$ . Navyiac z dôsledku 2.1.1.1 vieme, že vrchol  $v$  má práve dvoch susedov. Jeho druhého suseda si preto označme  $w$ .

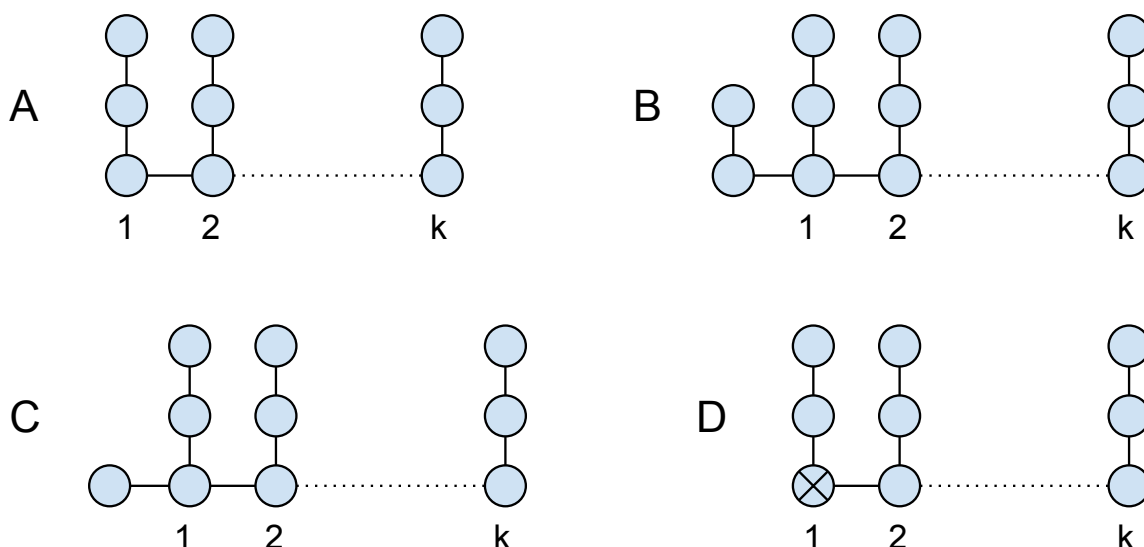
Ak je stupeň vrchola  $w$  najviac 2, vlastné páry si rozdelíme podľa toho, či ich 2-pakovania obsahujú vrchol  $u$ . Ak sa vrchol  $u$  nenachádza v 2-pakovaní, máme dve možnosti kam ho umiestniť a zvyšných  $n - 1$  môže byť vo vlastných pároch ľubovoľne. Pri druhej možnosti, už nemôžeme do 2-pakovania vložiť vrcholy  $v$  a  $w$ , tie majú 4 možnosti na umiestnenie a zvyšných  $n - 3$  vrcholov nám opäť nič neovplyvňuje. Vďaka tomu dostávame jednoduchú nerovnicu  $pp(n) \leq 2pp(n - 1) + 4pp(n - 3)$ .

V opačnom prípade však dostaneme oveľa zložitejší problém, v ktorom sa musíme pozerieť na všetky vrcholy grafu  $G$  vzdialené od vrchola  $w$  na vzdialenosť 1 a 2. Opäť sa však dá nájsť rekurzívne ohraničenie hodnoty  $pp(n)$ , aj keď v tomto prípade nemá taký jednoduchý tvar.

Z týchto odvodení sa ale dá ukázať, že  $pp(n) \leq 2\tau^n$ , pričom hodnota  $\tau$  je kladným koreňom rovnice  $\tau^5 = 16\tau + 88$ . Z tejto poslednej nerovnice vyplýva, že horný odhad na počet vlastných párov  $n$  vrcholového grafu je  $O(\tau^n) \approx O(2.6488^n)$ .

## 2.1.2 Dolný odhad počtu vlastných párov

Na odhadnutie počtu vlastných párov v grafoch prezentoval Junosza-Szaniawski a kolektív triedu grafov, ktorú môžeme vidieť na obrázku 2.1.



Obr. 2.1: Trieda grafov slúžiaca ako dolný odhad počtu vlastných párov.

Čísla  $a_k$ ,  $b_k$  a  $c_k$  budú označovať počet vlastných párov grafov  $A$ ,  $B$  a  $C$ . Číslo  $d_k$  je počet vlastných párov grafu  $D$ , v ktorých 2-pakovaní sa nenachádza preškrtnutý vrchol. Vďaka tomu vieme odvodiť nasledovné rekurencie.

$$\begin{aligned}
 a_k &= 2b_{k-1} + 4a_{k-1} \\
 b_k &= 2c_k + 2d_k \\
 c_k &= 2a_k + 12d_{k-1} \\
 d_k &= 4d_{k-1} + 12a_{k-1}
 \end{aligned} \tag{2.3}$$

Vyriešením tohto systému rovníc zistíme, že hodnota  $a_k = \Theta(x^k)$ , kde  $x$  je kladný koreň rovnice  $x^3 = 16x^2 + 576$ . Graf  $A$  preto obsahuje  $\Theta(17.8149\dots^{n/3}) = \Theta(2.6117^n)$  vlastných párov. Hodnota  $pp(n)$  je preto zospodu ohraničená hodnotou  $\Omega(2.6117^n)$ .

## 2.2 Stromy s najväčším počtom vlastných párov pre malý počet vrcholov

Prvá prirodzená otázka, ktorú si musíme položiť je, čomu sa rovná hodnota  $pp(n)$  pre malé čísla  $n$  a ako vyzerajú prislúchajúce stromy s najväčším počtom vlastných párov. Tak ako v mnohých kombinatorických problémoch sa môže stať, že z menších riešení sa dá odhadnúť všeobecný tvar výsledku. Poprípade nám to dá triedu stromov, ktorú budeme môcť použiť na dolný odhad.

Aby sme teda vedeli nájsť strom, ktorý má zaručene najviac vlastných párov, musíme vedieť robiť dve veci. Generovať všetky stromy s daným počtom vrcholov a vedieť pre tieto stromy vypočítať koľko vlastných párov obsahujú. Na oba tieto problémy sa pozrieme v nasledujúcich podkapitolách.

### 2.2.1 Generovanie všetkých neizomorfných stromov

Aby sme mohli zistiť, ktorý strom na  $n$  vrchoch má najväčší počet vlastných párov, musíme sa pozrieť na všetky nezakorenené stromy s  $n$  neoznačenými vrcholmi. Pri vlastných pároch totiž nezohráva úlohu ani označenie ani zakorenenie stromov. No napriek tomu, že takýchto navzájom neizomorfných stromov je menej ako pri iných alternatívach, zistíme, že pre  $n = 25$  ich je vyše sto miliónov a pre  $n = 28$  ich počet presahuje dve miliardy. Je preto jasné, že sa nám nepodarí vygenerovať všetky stromy pre príliš väčšie počty vrcholov.

Na samotné generovanie neizomorfných stromov sme sa rozhodli použiť už existujúcu implementáciu `nauty-geng`[11], ktorá slúži na generovanie všetkých neizomorfných grafov s požadovanými vlastnosťami, ako je napríklad počet vrcholov, hrán alebo súvislosť.

Táto knižnica síce nie je špecializovaná na generovanie stromov, nepodarilo sa nám však nájsť žiadne špecializovanejšie nástroje alebo už vygenerované zoznamy[1]. Navyiac, táto knižnica zvládla vygenerovať stromy až do veľkosti  $n = 23$ , čo bolo pre naše účely postačujúce.

Spomínanú implementáciu sme používali s nasledovnými parametrami:

```
for x in {1..23}; do
  nauty-geng $x $((($x-1)):$((($x-1))) -c -q
done
```

Nakoniec sme ešte museli vyriešiť výstupný formát, v ktorom boli grafy udávané. Kvôli komprimácii totiž implementácia `nauty-geng` používala formát `graph6`. Tento formát slúži na komprimovaný zápis neorientovaných grafov, ktoré majú menej ako  $2^{36}$  vrcholov. Funguje tak, že si zoberie horný trojuholník matice susednosti ako bitový vektor, ten rozdelí na šesticice reprezentujúce binárne čísla, ku ktorým pripočíta 63, čím dostane znaky z ASCII tabuľky z rozsahu 63 – 126. Tieto ASCII znaky následne používa na zapísanie daného grafu. Na jeho parsovanie do čitateľnejšieho zápisu, ktorý mohli používať naše algoritmy sme použili implementáciu `showg.c`[11].

Dodáme, že k dispozícii bol aj formát `sparse6`, ktorý namiesto matice susedností používa zoznam hrán a je preto vhodnejší pre riedke grafy, akými sú napríklad stromy. Keďže však generované grafy majú veľmi málo vrcholov, formát `graph6` je predsa len lepší.

### 2.2.2 Počítanie vlastných párov v strome

Keď už vieme vygenerovať všetky stromy s  $n$  vrcholmi, potrebujeme vedieť zistiť, ktorý z nich má najväčší počet vlastných párov. Pre každý z nich preto tento počet musíme vypočítať. Skôr ako si však ukážeme spôsob ako túto hodnotu spočítať, dokážeme nasledujúcu lemmu.

**Lemma 2.2.1.** *Nech  $G$  je graf a  $PP(G)$  je množina všetkých vlastných párov tohto grafu. Nech  $A$  a  $B$  sú disjunktné množiny vrcholov grafu  $G$ , také, že  $A \cup B = V(G)$ . Ak  $PP(A)$  je množina vlastných párov grafu indukovaného množinou  $A$  a  $PP(B)$  je množina vlastných párov grafu indukovaného množinou  $B$ , tak platí*

$$PP(G) = \{(S_1 \cup S_2, X_1 \cup X_2) \mid (S_1, X_1) \in PP(A), (S_2, X_2) \in PP(B), \\ S_1 \cup S_2 \text{ je 2-pakovanie grafu } G\} \quad (2.4)$$

*Dôkaz.* Je jasné, že každá dvojica  $(S_1 \cup S_2, X_1 \cup X_2)$  spĺňajúca uvedené vlastnosti je vlastným párom grafu  $G$ . No a každý vlastný pár  $(S, X)$  grafu  $G$  vieme rozdeliť na dva vlastné páry  $(S \cap A, X \cap A)$  a  $(S \cap B, X \cap B)$ , ktoré sú vlastnými párami v príslušných indukovaných podgrafoch.  $\square$

Najpriamočiarejšie riešenie bude postupne pre každú podmnožinu vrcholov overovať, či je 2-pakovaním. A ak poznáme veľkosti všetkých 2-pakovaní, vieme počet vlastných párov vypočítať pomocou vzorca 2.0.1. Takéto riešenie má časovú zložitosť  $O(n^2 \cdot 2^n)$ . Uvedomme si však, že tento algoritmus budeme musieť spúšťať pre každý vygenerovaný strom a tých bolo pre  $n = 23$  niekoľko desiatok miliónov. Preto nebude možné takýto prístup efektívne využiť.

Ak chceme počítať počet vlastných párov pre značné množstvo stromov, budeme musieť prísť s rýchlejšim riešením. Na to využijeme princíp dynamického programovania. Majme strom  $G$ , ktorý zakoreníme za ľubovoľný vrchol. Definujeme si nasledovné tri hodnoty:

- $P(v, 0)$  – počet vlastných párov  $(S, X)$  podstromu  $G$  s koreňom vo vrchole  $v$ , pričom vrchol  $v$  leží v 2-pakovaní  $S$



- $P(v, 1)$  – počet vlastných párov  $(S, X)$  podstromu  $G$  s koreňom vo vrchole  $v$ , pričom práve jeden syn vrchola  $v$  leží v 2-pakovaní  $S$
- $P(v, 2)$  – počet vlastných párov  $(S, X)$  podstromu  $G$  s koreňom vo vrchole  $v$ , pričom v 2-pakovaní  $S$  neleží ani vrchol  $v$  ani žiaden jeho syn

Každý takejto hodnote prirodzene prislúcha aj množina obsahujúca dané vlastné páry.

**Veta 2.2.2.** *Nech  $G$  je strom zakorenený vo vrchol  $w$ . Potom platí, že*

$$pp(G) = P(w, 0) + P(w, 1) + P(w, 2) \quad (2.5)$$

*Dôkaz.* Uvedomme si, že ak syn koreňa  $w$  leží v 2-pakovaní, tak v ňom nemôže ležať samotný vrchol  $w$ , lebo takáto množina by nebola 2-pakovaním. Takisto nemôžu v 2-pakovaní ležať viacerí synovia koreňa  $w$ .

Z toho vyplýva, že každý vlastný pár  $(S, X)$  grafu  $G$  je jednoznačne zaraditeľný do jednej z vymenovaných množín podľa toho, či vrchol  $w$  a jeho synovia ležia v  $S$ . No a hodnoty  $P(w, x)$  popisujú vlastné páry grafu  $G$  priamo z definície.  $\square$

Ostáva nám už len zistiť, ako rekurzívne vypočítať hodnoty  $P(w, 0)$ ,  $P(w, 1)$  a  $P(w, 2)$ . To nám ukážu nasledujúce tri lemmy.

**Lemma 2.2.3.** *Nech  $v$  je vrchol zakoreneného stromu  $G$ . Označme si jeho synov  $v_1, v_2 \dots v_k$ . Potom platí, že*

$$P(v, 0) = \prod_{i=1}^k P(v_i, 2) \quad (2.6)$$

*Dôkaz.* Na dokazovanie použijeme matematickú indukciu vzhľadom na počet synov vrchola  $v$ . Ak vrchol  $v$  nemá žiadneho syna, tento vrchol tvorí celý príslušný podstrom. A keďže z definície hodnoty  $P(v, 0)$  musí tento vrchol ležať v 2-pakovaní, máme práve jeden možný vlastný pár  $(\{v\}, \emptyset)$ . To zodpovedá hodnote prázdneho súčinu.

Následne predpokladajme, že platí indukčný predpoklad pre menej ako  $k$  synov. Podstrom tvorený vrcholom  $v$  a synmi  $v_1$  až  $v_{k-1}$ , ktorý si označíme  $G'$ , má preto  $p(G') = \prod_{i=1}^{k-1} P(v_i, 2)$  vlastných párov, pričom v 2-pakovaní každého tohto vlastného páru sa nachádza vrchol  $v$ . Ako sme si ukázali v lemme 2.2.1, aby sme mohli spájať vlastné páry stromu  $G'$  a podstromu pod vrcholom  $v_k$ , musí byť zjednotenie ich 2-pakovaní validné 2-pakovanie. V 2-pakovaní vlastného páru pre podstrom  $v_k$  sa teda nemôže nachádzať ani vrchol  $v_k$ , ani žiaden jeho syn. Tomu ale zodpovedá práve hodnota  $P(v_k, 2)$ . Prinásobením tejto hodnoty k  $p(G')$  dostaneme požadovaný výsledok.  $\square$

**Lemma 2.2.4.** *Nech  $v$  je vrchol zakoreneného stromu  $G$ . Označme si jeho synov  $v_1, v_2 \dots v_k$ . Potom platí, že*

$$P(v, 1) = 2 \sum_{i=1}^k \left( P(v_i, 0) \cdot \prod_{1 \leq j \leq k, j \neq i} (P(v_j, 1) + P(v_j, 2)) \right) \quad (2.7)$$

*Dôkaz.* Opäť použijeme matematickú indukciu. Ak nemá vrchol  $v$  ani jedného syna, hodnota prázdnej sumy je 0. Avšak, neexistuje ani žiaden vlastný pár, keďže z definície sa v jeho 2-pakovaní musí nachádzať nejaký syn vrchola  $v$ .

Predpokladajme, že pre podstrom s  $k-1$  synmi  $v_1$  až  $v_{k-1}$ , ktorý si označíme  $G'$ , daná lemma platí, teda  $p(G') = 2 \sum_{i=1}^{k-1} (P(v_i, 0) \prod_{1 \leq j \leq k-1, j \neq i} (P(v_j, 1) + P(v_j, 2)))$ . Opäť sa pozrieme, ako vyzerajú vlastné páry, keď k stromu  $G'$  pridáme aj podstrom pod vrcholom  $v_k$ .

Tak ako predtým, použijeme lemmu 2.2.1. Vo vlastnom páre  $G'$  sa v 2-pakovaní nachádza práve jeden syn vrchola  $v$ . Preto sa v 2-pakovaní podstromu  $v_k$  nemôže nachádzať jeho koreň, lebo vzájomná vzdialenosť dvoch synov vrchola  $v$  je 2. Môžeme však kombinovať vlastné páry podstromu  $v_k$ , ktoré majú v 2-pakovaní niektorého zo synov vrchola  $v_k$  alebo tie, ktoré nemajú v 2-pakovaní ani vrchol  $v_k$  ani žiadneho jeho syna. To nám dá  $p(G') \cdot (P(v_k, 1) + P(v_k, 2))$  možností.

Musíme si však uvedomiť, že to nepokrýva možnosti, keď sa v 2-pakovaní podstromu  $v$  nachádza vrchol  $v_k$ . Musíme preto kombinovať vlastné páry podstromu  $v_k$ , v ktorých sa  $v_k$  nachádza v 2-pakovaní a stromu  $G'$ , v ktorých 2-pakovaní neleží ani vrchol  $v$  ani žiaden jeho syn. Počet takýchto vlastných párov je  $2 \prod_{i=1}^{k-1} (P(v_i, 1) + P(v_i, 2))$ . Dvojka na začiatku je započítaná za vrchol  $v$ , ktorý sa buď nachádza alebo nenachádza v druhej množine vlastného páru. Zvyšok dostaneme ako počty vlastných párov pre jednotlivé podstromy, ak v ich 2-pakovaní nemôže byť ich koreň.

Výsledný počet vlastných párov je preto

$$p(G') \cdot (P(v_k, 1) + P(v_k, 2)) + 2 \cdot P(v_k, 0) \cdot \prod_{i=1}^{k-1} (P(v_i, 1) + P(v_i, 2))$$

čo vieme upraviť na požadovaný tvar. □

**Lemma 2.2.5.** *Nech  $v$  je vrchol zakoreneného stromu  $G$ . Označme si jeho synov  $v_1, v_2 \dots v_k$ . Potom platí, že*

$$P(v, 2) = 2 \prod_{i=1}^k (P(v_i, 1) + P(v_i, 2)) \quad (2.8)$$

*Dôkaz.* Použijeme matematickú indukciu. Ak vrchol  $v$  nemá ani jedného syna, existujú práve dva prípustné vlastné páry –  $(\emptyset, \{v\})$  a  $(\emptyset, \emptyset)$ . Ich počet teda zodpovedá dvojnásobku prázdneho súčinu.

Ako indukčný predpoklad si stanovme počet vlastných párov pre strom  $G'$  skladajúci sa z vrchola  $v$  a z podstromov synov  $v_1$  až  $v_k$ . Platí, že  $p(G') = 2 \prod_{i=1}^{k-1} (P(v_i, 1) + P(v_i, 2))$ . Pre tieto vlastné páry si navyše uvedomme, že ľubovoľný vrchol  $v$  ich 2-pakovaní je od vrchola  $v$  vzdialený aspoň o 2 a preto sú dostatočne vzdialené od všetkých vrcholov podstromu  $v_k$ .

Vlastné páry podstromu  $v_k$  však nemôžu v 2-pakovaní obsahovať vrchol  $v_k$ , keďže by to nespĺňalo definíciu hodnoty  $P(v, 2)$ . Všetky zvyšné vlastné páry, ktorých je  $P(v_k, 1) + P(v_k, 2)$  sú však prípustné. Kombináciou vlastných párov pre strom  $G'$  a podstrom  $v_k$  preto dostaneme  $p(G') \cdot (P(v_k, 1) + P(v_k, 2))$  rôznych vlastných párov, čo vieme upraviť do požadovaného tvaru.  $\square$

Vyššie uvedené lemy 2.2.3, 2.2.4 a 2.2.5 nám hovoria, ako vypočítať hodnoty  $P(v, x)$  rekurzívne v čase úmernom od počtu synov vrchola  $v$ . Použitím princípu memoizácie, v ktorom si budeme pamätať už vypočítané rekurzívne volania, dokážeme implementovať algoritmus, ktorý zistí počet vlastných párov stromu  $G$  s  $n$  vrcholmi pomocou  $\Theta(n)$  aritmetických operácií.

Dodáme, že napriek tomu že sa na prvý pohľad zdá, že na počítanie hodnoty  $P(w, 1)$  potrebujeme až  $\Theta(n^2)$  aritmetických operácií, v ďalšej časti si ukážeme šikovný trik, ako túto zložitosť zredukovať.

### 2.2.3 Implementácia

Pri samotnej implementácii sme sa museli popasovať s viacerými problémami. V prvom rade bola dôležitá efektivita navrhovaného algoritmu, keďže sme ho potrebovali spúšťať na veľkom množstve rôznych stromov. Aj z tohto dôvodu sme sa rozhodli ho implementovať v jazyku C++, ktorý je oproti alternatívam ako Python výrazne rýchlejší.

Už spomínaním problémom bol výpočet hodnôt  $P(v, 1)$  v čase lineárnom od počtu jeho synov. To sme však vedeli vyriešiť tak, že sme si najskôr spočítali hodnotu  $T = \prod_{i=1}^k (P(v_i, 1) + P(v_i, 2))$ , vďaka čomu sme mohli použiť vzorec:

$$P(v, 1) = 2 \sum_{i=1}^k \frac{P(v_i, 0) \cdot T}{P(v_i, 1) + P(v_i, 2)} \quad (2.9)$$

Ďalším problémom sa ukázal byť samotný počet vlastných párov stromu  $G$ . Ak má strom  $G$   $n$  vrcholov, počet vlastných párov sa pohybuje medzi  $2^n$  a  $3^n$ . Pre rastúce  $n$  začne počet vlastných párov stromu presahovať veľkosť 64-bitovej premennej `long long int`. Síce sa to nestane pre grafy do veľkosti 23, neskôr sa nám však zide poznať počet vlastných párov aj pre zopár väčších stromov.

Z tohto dôvodu sme museli implementovať vlastnú triedu pracujúcu s veľkými číslami. Implementovať sčítanie a takisto násobenie bolo jednoduché a neovplyvnilo príliš reálny čas výpočtu algoritmu, napriek tomu, že sme násobenie neimplementovali optimálnym spôsobom. Vo vzorci 2.9 však potrebujeme delenie dvoch veľkým čísel, ktorého vlastná implementácia je komplikovaná a pomalá.

Aby sme sa vyhli deleniu veľkých čísel, implementovali sme rátanie hodnoty  $P(v, 1)$  iným spôsobom. Najskôr sme si na  $i$ -tu pozíciu poľa `A[]` uložili hodnoty  $P(v_i, 1) + P(v_i, 2)$ . Potom sme si z poľa `A[]` vypočítali pole prefixových súčinov `Pref[]` a pole sufixových súčinov `Suf[]`. Hodnotu  $P(v, 1)$  sme potom počítali ako

$$P(v, 1) = 2 \sum_{i=1}^k P(v_i, 0) \cdot Pref[i - 1] \cdot Suf[i + 1] \quad (2.10)$$

Výsledný algoritmus potreboval na výpočet  $\Theta(n)$  aritmetických operácií. Keďže sme však museli pracovať s veľkými číslami, tieto operácie nevieme vykonávať v konštantnom čase. Veľkosť čísel, s ktorými sme pracovali, však nepresiahla  $3^n$ . Preto mali naše čísla najviac  $n$  cifier. Naša implementácia násobenia dvoch takto veľkých čísel potrebovala  $\Theta(n^2)$  operácií, z čoho dostávame výslednú zložitosť  $\Theta(n^3)$ .

Na záver dodáme, že ak by sme na násobenie použili rýchlejší algoritmus, napríklad rýchle Fourierove násobenie, vedeli by sme dosiahnuť časovú zložitosť až  $\Theta(n^2 \log(n) \log(\log(n)))$ . Vzhľadom na veľkosť dát, s ktorými sme pracovali to však nebolo potrebné.

Nasledujúci kus programu ukazuje implementáciu rekurzívnej funkcie na rátanie hodnôt  $P(v, x)$  za použitia memoizácie bez vlastnej implementácie veľkých čísel.

### Listing programu (C++)

```
int n;
vector<vector<int> > Graph;
vector<vector<long long int> > Mem;

long long int count_proper_pairs(int v, int parent, int type) {
    if(Mem[v][type] != -1) return Mem[v][type];
    int neig = 0;
    //vypocitaj pocet susedov
    for (int i = 0; i < Graph[v].size(); i++) {
        int w = Graph[v][i];
        if(w != parent) neig++;
    }

    if(type == 0) {
        Mem[v][type] = 1;
        for(int i = 0; i < Graph[v].size(); i++) {
            int w = Graph[v][i];
            if(w == parent) continue;
            Mem[v][type] *= count_proper_pairs(w, v, 2);
        }
    }

    if(type == 1) {
        Mem[v][type] = 0;
        long long int pom = 1;
        for(int i = 0; i < Graph[v].size(); i++) {
            int w = Graph[v][i];
            if(w == parent) continue;
            pom *= (count_proper_pairs(w, v, 1) + count_proper_pairs(w, v, 2));
        }
        for(int i = 0; i < Graph[v].size(); i++) {
            int w = Graph[v][i];
            if(w == parent) continue;
            Mem[v][type] += count_proper_pairs(w, v, 0)
                * pom / (count_proper_pairs(w, v, 1) + count_proper_pairs(w, v, 2));
        }
        Mem[v][type] *= 2;
    }

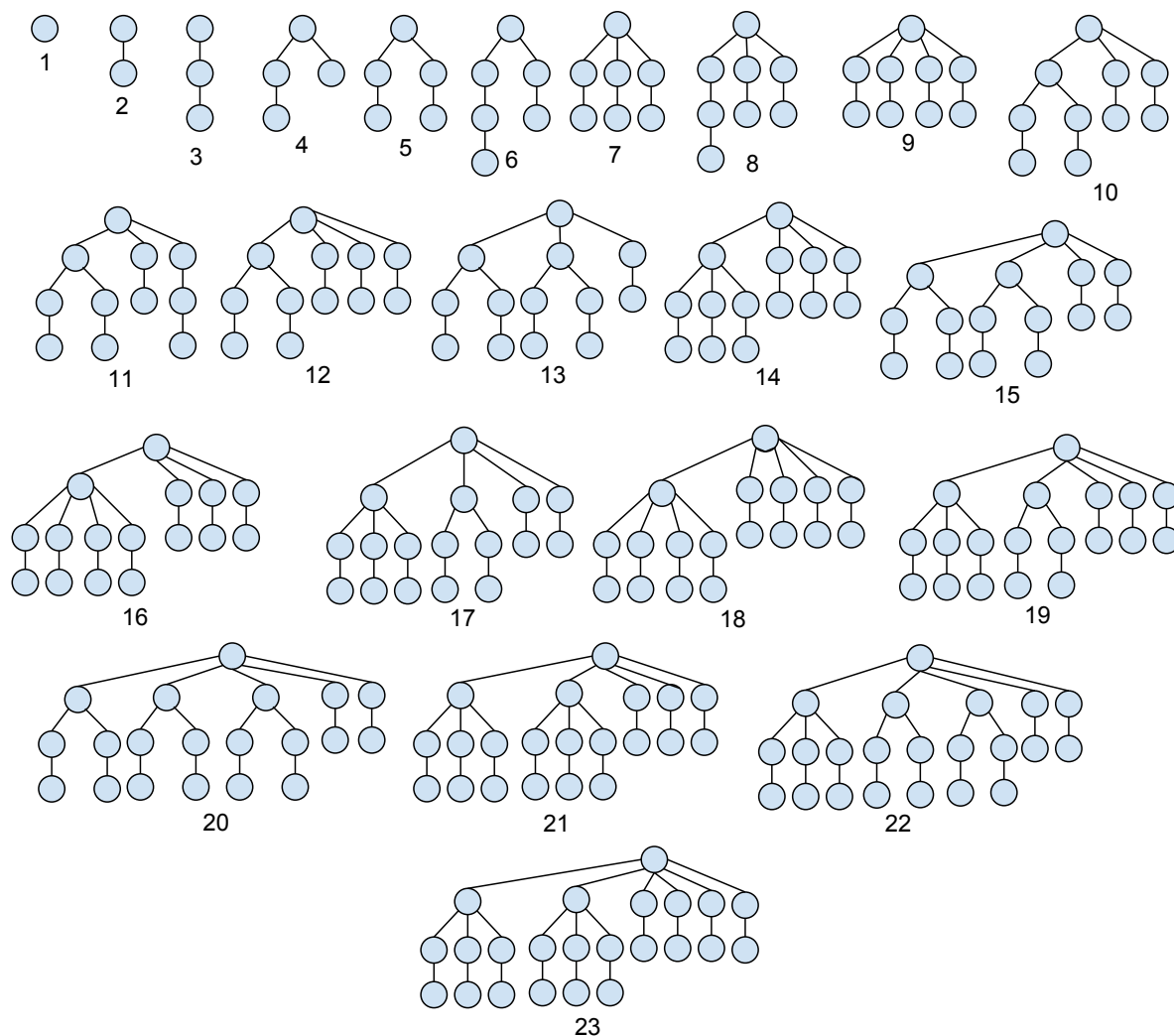
    if(type == 2) {
        Mem[v][type] = 1;
        for(int i = 0; i < Graph[v].size(); i++) {
            int w = Graph[v][i];
            if(w == parent) continue;
            Mem[v][type] *= (count_proper_pairs(w, v, 2) + count_proper_pairs(w, v, 1));
        }
        Mem[v][type] *= 2;
    }

    return Mem[v][type];
}
```

## 2.3 Experimenty na stromoch s malým počtom vrcholov

V predchádzajúcej časti sme si predstavili algoritmus, ktorý generuje všetky stromy s  $n$  vrcholmi a potom pre ne zisťuje, ktorý z nich má najväčší počet vlastných párov. Tento algoritmus sme v rámci našej práce skutočne implementovali a spustili. Ako

sme spomínali v predchádzajúcich častiach, bol schopný prezrieť všetky stromy, ktoré mali najviac 23 vrcholov a pre každý počet vrcholov vybrať ten s najväčším počtom vlastných párov. Tieto najhoršie stromy boli takmer vždy jednoznačné, s výnimkou prípadov  $n = 11$ ,  $n = 13$  a  $n = 23$ , kde boli dve neizomorfné možnosti s rovnakým počtom vlastných párov. Ako vyzerajú spomínané stromy si môžete pozrieť v nasledujúcom obrázku 2.2.



Obr. 2.2: Stromy s najväčším počtom vlastných párov s daným počtom vrcholov.

Stromy na obrázku sú kreslené zakorenené. Toto zakorenenie nie je náhodné, slúži na zvýraznenie štruktúry, ktorú sme si všimli pri analýze dát. Ako si môžeme všimnúť, každý z nakreslených stromov má jeden koreň, pod ktorým visí niekoľko rôzne veľkých podstromov. Dôležité na tom však je, že ak podstrom obsahuje  $k$  vrcholov, tak má tvar stromu veľkosti  $k$  s najväčším počtom vlastných párov.

Naskytá sa preto otázka, či stromy s najväčším počtom vlastných párov nemajú takúto rekurzívnu štruktúru aj vo všeobecnosti. Zaved'eme nasledovnú definíciu.

**Definícia 2.3.1.** *Nech  $G$  je strom s  $n$  vrcholmi zakorenený vo vrchole  $v$ . Podstromy visiace pod koreňom  $v$  majú veľkosti  $n_1, n_2 \dots n_k$ . Strom  $G$  nazveme **rekurzívny**, ak pre všetky  $i$  platí, že  $i$ -ty podstrom koreňa je rekurzívny strom na  $n_i$  vrcholoch s najväčším počtom vlastných párov.*

Radi by sme sa teraz pozreli na rekurzívne stromy z definície 2.3.1. Výhodou je, že rekurzívnych stromov s  $n$  vrcholmi je oveľa menej ako všetkých stromov s  $n$  vrcholmi. Presnejšie, počet rekurzívnych  $n$  vrcholových stromov je rovný počtu rozdelení čísla  $n - 1$  na množinu sčítancov, ktoré určujú veľkosti jednotlivých podstromov pod koreňom. Počet takýchto rozdelení však rastie oveľa pomalšie ako počet všetkých stromov. Napríklad, pre  $n = 23$  máme iba 1001 možných rozdelení a hodnoty väčšie ako milión sa dosahujú na rádovo 70-tich vrcholoch.

Vďaka tomu, budeme vedieť získať rekurzívne stromy s najväčším počtom vlastných párov pre oveľa väčšie hodnoty  $n$ . Nevýhodou však je, že stratíme na všeobecnosti. Pri takomto obmedzení už nebudeme môcť tvrdiť, že náš algoritmus nájde všeobecný strom s najväčším počtom vlastných párov. To, v čo môžeme dúfať, je nájdenie triedy stromov, ktorá poskytuje lepší dolný odhad a hlbšie pochopenie riešeného problému.

Z týchto dôvodov sme implementovali algoritmus, ktorý vytváral rekurzívne stromy a použitím rovnakého dynamického programovania, aké sme použili na všeobecný prípad, sme medzi nimi hľadali tie s najväčším počtom vlastných párov. Pomocou neho sa nám podarilo nájsť najhoršie rekurzívne stromy pre stromy do veľkosti 70 vrcholov. V dátach, ktoré sme takýmto spôsobom dostali sa následne môžeme pozeráť na dve vlastnosti. Po prvé, na počet vlastných párov týchto maximálnych rekurzívnych stromov a po druhé na štruktúralne vlastnosti týchto stromov.

Vieme, že počet vlastných párov budeme chcieť odhadnúť ako  $\Theta(c^n)$ . Preto nás zaujíma, aká by musela byť hodnota  $c$ , aby sa  $c^n$  rovnalo počtu vlastných párov najhoršieho rekurzívneho stromu na  $n$  vrcholoch. Ak má táto trieda stromov slúžiť ako lepší dolný odhad na počet vlastných párov, dá sa predpokladať, že hodnota  $c$  bude konvergovať k výslednému dolnému odhadu. No a ak by hodnota  $c$  bola menšia ako už dokázaný dolný odhad, mohli by sme usúdiť, že táto cesta uvažovania nie je správna.

V tabuľke 2.1 si môžeme všimnúť, že nami hľadaná hodnota  $c$  sa pohybuje okolo hodnoty 2.63. Takisto výkyvy od tejto hodnoty sú pomerne malé, aj keď nie znižujúce sa.

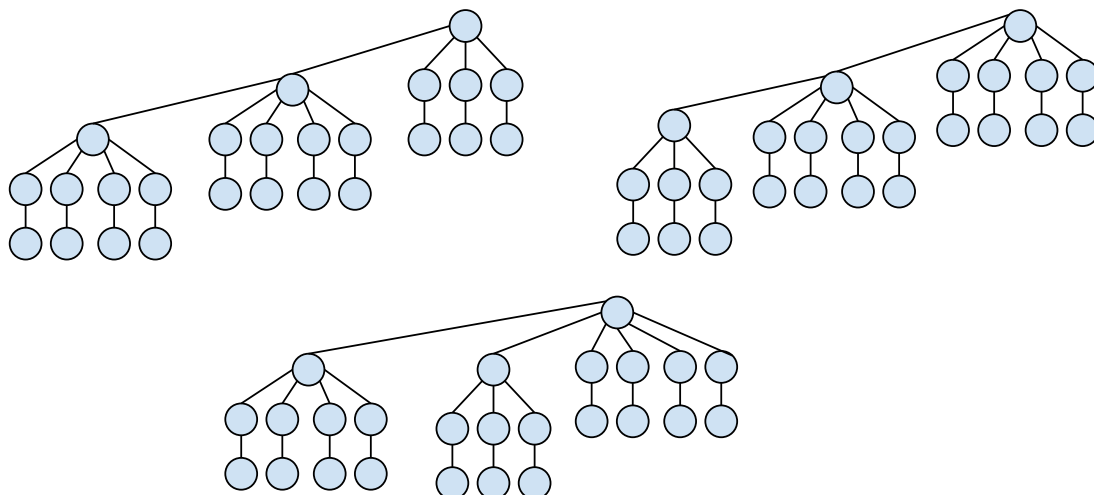
$n$	počet vlastných párov	hodnota $c$	absolútna zmena hodnoty $c$
1	3	3.000000000	-
2	8	2.828427125	0.171572875
3	20	2.714417617	0.114009508
4	52	2.685349614	0.029068002
5	136	2.671208461	0.014141153
6	352	2.657197146	0.014011315
⋮	⋮	⋮	⋮
64	921187831485255720716206080	2.638262111	0.000050569
65	2406306422819565242004013056	2.637858848	0.000403263
66	6441340731750529031831814144	2.638445485	0.000586637
67	16786859959489808408883757056	2.637959970	0.000485515
68	45083096797426091251753549824	2.638654662	0.000694692
69	117680487160328897201770070016	2.638241559	0.000413103
70	311280463399896050550976806912	2.638339873	0.000098313

Tabuľka 2.1: Počet vlastných párov v rekurzívnych stromoch

Zostáva nám pozrieť sa na to, ako vyzerajú nájdené rekurzívne stromy. Hneď na začiatku si všimneme, že takmer každý strom má viacero alternatív, ako môže vyzeráť. Rýchlo však nahliadneme, že tieto stromy sú navzájom izomorfné (až na spomenuté výnimky 11, 13 a 23). Napríklad strom s 25 vrcholmi má 3 možné rekurzívne zakreslenia, ktoré môžete vidieť na obrázku 2.3, líšia sa len tým, ktorí z vrcholov je koreňom tohto stromu.

Ak máme viacero alternatív, ako zakresliť rekurzívny strom na  $n$  vrcholoch, vyberieme si tú, ktorá minimalizuje veľkosť najväčšieho podstromu. Náš algoritmus však bude naďalej pracovať so všetkými alternatívami, lebo umiestnenie koreňa podstromu





Obr. 2.3: Tri možné zakreslenia rekurzívneho stromu s 25 vrcholmi. V prvom prípade má podstrom veľkosti 18 a tri podstromy veľkosti 2, v druhom prípade má podstrom veľkosti 16 a štyri podstromy veľkosti 2 a v poslednom prípade má podstromy veľkostí 9, 7 a štyri podstromy veľkosti 2.

môže ovplyvniť výsledný počet vlastných párov. V tabuľke 2.2 si môžeme prezrieť nájdené rozloženia maximálnych stromov. Môžeme si v nich všimnúť, že žiaden podstrom nie je väčší ako 9 vrcholov. No a bližším preskúmaním dát zistíme, že priemer žiadneho rekurzívneho stromu s najväčším počtom vlastných párov nemá priemer väčší ako 6.

Naskytajú sa preto otázka, ako vyzerajú stromy s priemerom 6 a či to nie sú práve nami hľadané stromy s najväčším počtom vlastných párov. Ak nič iné, mohli by nám dať aspoň dobre uchopiteľnú triedu stromov, určujúcu dolný odhad. A vďaka malému priemeru je ich štruktúra pomerne limitovaná, vďaka čomu by sa nám mohlo podariť dokázať o nich aj iné vlastnosti. Preto sa v nasledujúcej kapitole pozrieme práve na stromy s priemerom 6.

$n$	veľkosti podstromov	$n$	veľkosti podstromov	$n$	veľkosti podstromov
1	-	25	9, 7, 2, 2, 2, 2	50	7, 7, 7, 7, 7, 5, 5, 2, 2
2	1	26	7, 7, 5, 2, 2, 2	51	9, 7, 7, 7, 7, 7, 2, 2, 2
3	2	27	7, 5, 5, 5, 2, 2	52	7, 7, 7, 7, 7, 7, 5, 2, 2
4	2, 1	28	7, 7, 7, 2, 2, 2	53	7, 7, 7, 7, 7, 5, 5, 5, 2
5	2, 2	29	7, 7, 5, 5, 2, 2	54	7, 7, 7, 7, 7, 7, 7, 2, 2
6	3, 2	30	9, 7, 7, 2, 2, 2	55	7, 7, 7, 7, 7, 7, 5, 5, 2
7	2, 2, 2	31	7, 7, 7, 5, 2, 2	56	7, 7, 7, 7, 7, 7, 7, 2, 2, 2
8	3, 2, 2	32	9, 7, 7, 2, 2, 2, 2	57	7, 7, 7, 7, 7, 7, 7, 5, 2
9	2, 2, 2, 2	33	7, 7, 7, 7, 2, 2	58	9, 7, 7, 7, 7, 7, 7, 2, 2, 2
10	5, 2, 2	34	7, 7, 5, 5, 5, 2, 2	59	7, 7, 7, 7, 7, 7, 7, 5, 2, 2
11	4, 2, 2, 2	35	7, 7, 7, 7, 2, 2, 2	60	7, 7, 7, 7, 7, 7, 5, 5, 5, 2
12	5, 2, 2, 2	36	7, 7, 7, 5, 5, 2, 2	61	7, 7, 7, 7, 7, 7, 7, 7, 2, 2
13	5, 3, 2, 2	37	9, 7, 7, 7, 2, 2, 2	62	7, 7, 7, 7, 7, 7, 7, 5, 5, 2
14	7, 2, 2, 2	38	7, 7, 7, 7, 5, 2, 2	63	9, 7, 7, 7, 7, 7, 7, 7, 2, 2
15	5, 5, 2, 2	39	9, 9, 7, 7, 2, 2, 2	64	7, 7, 7, 7, 7, 7, 7, 7, 5, 2
16	7, 2, 2, 2, 2	40	7, 7, 7, 7, 7, 2, 2	65	9, 9, 7, 7, 7, 7, 7, 7, 2, 2
17	7, 5, 2, 2	41	7, 7, 7, 5, 5, 5, 2, 2	66	7, 7, 7, 7, 7, 7, 7, 7, 7, 2
18	9, 2, 2, 2, 2	42	7, 7, 7, 7, 7, 2, 2, 2	67	7, 7, 7, 7, 7, 7, 7, 5, 5, 5, 2
19	7, 5, 2, 2, 2	43	7, 7, 7, 7, 5, 5, 2, 2	68	7, 7, 7, 7, 7, 7, 7, 7, 7, 2, 2
20	5, 5, 5, 2, 2	44	9, 7, 7, 7, 7, 2, 2, 2	69	7, 7, 7, 7, 7, 7, 7, 7, 5, 5, 2
21	7, 7, 2, 2, 2	45	7, 7, 7, 7, 7, 5, 2, 2	70	9, 7, 7, 7, 7, 7, 7, 7, 7, 2, 2
22	7, 5, 5, 2, 2	46	9, 9, 7, 7, 7, 2, 2, 2		
23	7, 7, 2, 2, 2, 2	47	7, 7, 7, 7, 7, 7, 2, 2		
24	7, 7, 5, 2, 2	48	7, 7, 7, 7, 7, 5, 5, 2		

Tabuľka 2.2: Rozloženie rekurzívnych stromov na podstromy minimalizujúce veľkosť najväčšieho podstromu

# Kapitola 3

## Vlastné páry na stromoch s priemerom 6

V predchádzajúcej kapitole sme si ukázali, že všetky stromy s najväčším počtom vlastných párov, ktoré sme vygenerovali mali priemer najviac 6. Preto nás zaujímalo, či dokážeme určiť, koľko vlastných párov majú takéto stromy a či nevieme dokázať, že tieto stromy majú najväčší počet vlastných párov aj vo všeobecnosti.

V tejto kapitole si preto ukážeme, ako vyzerajú stromy s priemerom 6, stanovíme dolný odhad na počet vlastných párov  $n$  vrcholového stromu s priemerom 6 a ukážeme, že tento odhad je asymptoticky presný odhad maximálneho počtu vlastných párov. Navyše spravíme algoritmus, ktorý bude vedieť nájsť  $n$  vrcholový strom s priemerom 6 s najväčším počtom vlastných párov pomocou  $O(\log n)$  aritmetických operácií.

### 3.1 Počet vlastných párov v stromoch s priemerom 6

#### 3.1.1 Tvar stromov s priemerom 6

Skôr ako sa pustíme do počítania počtu vlastných párov v stromoch s priemerom 6, pozrieme sa na to, ako tieto stromy vlastne vyzerajú. Tým, že ich priemer je pomerne malý je veľmi obmedzený aj ich samotný tvar. Ak si zoberieme ľubovoľnú najdlhšiu cestu, vyskytne sa na nej 7 vrcholov. Vieme teda jednoznačne určiť stredný vrchol, za ktorý zakoreníme náš strom. Tento vrchol je navyše zdieľaný všetkými najdlhšími cestami, takže jeho výber je jednoznačný.

Každý strom s priemerom 6 si teda môžeme zakoreniť za tento stredný vrchol, pod ktorým bude ležať niekoľko podstromov, každý s hĺbkou najviac 2. Každý takýto strom sa preto skladá zo štyroch úrovní očíslovaných 0 až 3. Číslo úrovne určuje vzdialenosť

od koreňa stromu. To znamená, že na úrovni 0 sa nachádza iba koreň, na úrovni 1 jeho synovia atď.. K ďalším záverom o tvare týchto stromov budeme potrebovať tvrdenie 2.1.1.1, ktoré ukázalo, že žiadne dva listy nemajú spoločného suseda.

Keď teda hľadáme stromy s priemerom 6, ktoré majú najväčší počet vlastných párov, môžeme sa pozerať na tie, v ktorých žiadny vrchol nemá dvoch susedov, ktorý sú zároveň listami. Ako uvidíme, táto vlastnosť nám výrazne obmedzí množinu skúmaných grafov a zjednoduší prácu s nimi.

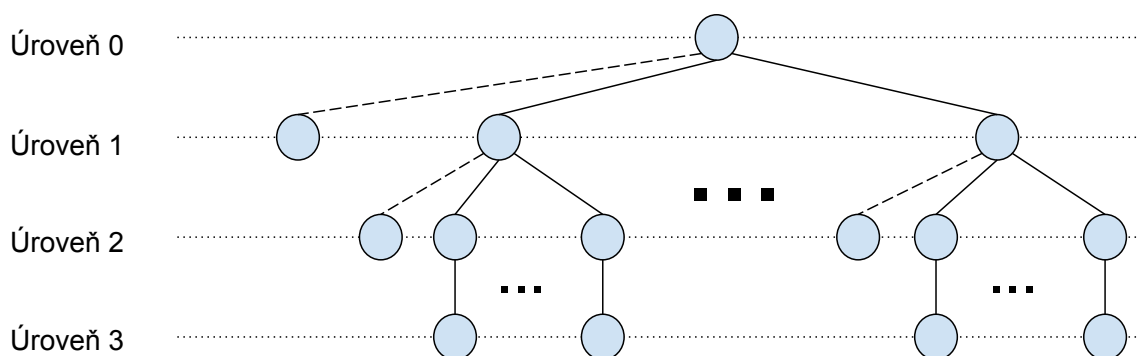
Ako sme spomínali, každý strom s priemerom 6 vieme zakoreniť za jednoznačný vrchol a následne sa jeho vrcholy rozdelia na 4 nezávislé úrovne. Z dôsledku 2.1.1.1 nám potom vyplýva, že každý vrchol môže mať najviac jedného suseda, ktorý je zároveň listom. Ďalej si uvedomme, že ak má vrchol  $v$  suseda  $u$  a vrchol  $u$  je list, tak tento vrchol leží na o jedno vyššej úrovni ako vrchol  $v$ . Táto vlastnosť platí aj pre koreň a jeho susedov, keďže koreň nemôže byť list.

Naviac vieme, že všetky vrcholy na úrovni 3 (tej najspodnejšej) sú listy, pretože nemajú žiadnych susedov na nižšej úrovni. Tým pádom ale vrcholy na úrovni 2 nemôžu mať viac ako jedného syna na úrovni 3, lebo by sme narušili vlastnosť z dôsledku 2.1.1.1.

Strom s priemerom 6, ktorý spĺňa všetky vyššie uvedené vlastnosti musí preto vyzeráť nasledovne:

- Na úrovni 0 je práve jeden vrchol (koreň), ktorý môže mať ľubovoľný počet synov. Najviac jeden z týchto synov je list.
- Na úrovni 1 sú všetci synovia koreňa. Každý z týchto vrcholov môže mať ľubovoľný počet synov. Najviac jeden z týchto synov je list.
- Na úrovni 2 sú všetci synovia vrcholov z úrovne 1. Každý z týchto vrcholov má najviac jedného syna.
- Na úrovni 3 sú všetci synovia vrcholov z úrovne 2. Tieto vrcholy nemajú žiadnych ďalších synov.

Všetky uvedené vlastnosti sú ilustrované na obrázku 3.1.



Obr. 3.1: Všeobecný tvar stromu s priemerom 6, ktorý má najväčší počet vlastných párov.

Uvedomme si, že tvar podstromu zaveseného za koreň takéhoto stromu je jednoznačne určený počtom jeho vrcholov. Vezmime si totiž koreň tohto podstromu. Pod ním je zavesený najviac jeden list a potom niekoľko dvojíc vrcholov, ktoré tvoria úroveň 2 a 3. Ak má náš podstrom  $n$  vrcholov, tak ak je číslo  $n$  nepárne, je tvorený koreňom a  $\frac{n-1}{2}$  dvojicami vrcholov a ak je číslo  $n$  párne, tak je tvorený koreňom, jedným listom zaveseným za koreň a  $\frac{n-2}{2}$  dvojicami vrcholov.

**Definícia 3.1.1.** *Nech  $G$  je  $n$  vrcholový zakorenený strom s hĺbkou najviac 2, v ktorom žiaden vrchol nemá viac ako jedného syna, ktorý je zároveň listom. Takýto strom voláme pravidelný strom veľkosti  $n$ .*

Naše stromy s priemerom 6 sa preto skladajú z niekoľkých pravidelných stromov, ktoré sú koreňmi pripojené k jednému spoločnému vrcholu. Navyiac, ako sme si povedali, tieto pravidelné stromy sú určené jednoznačne ich veľkosťou. Strom s priemerom 6 je preto jednoznačne určený veľkosťou jednotlivých pravidelných stromov.

### 3.1.2 Rátanie počtu vlastných párov

V predchádzajúcej časti sme si ukázali, ako vieme jednoznačným spôsobom zapísať stromy s priemerom 6, ktoré budú mať najväčší počet vlastných párov. Zaujímalo by nás preto, ako pre ne vieme spočítať počet vlastných párov. Samozrejme, opäť môžeme použiť algoritmus z predchádzajúcej kapitoly. Keďže sú však tieto stromy pomerne špeciálne, mohol by existovať aj rýchlejší spôsob. Dokonca by sme nepotrebovali samotný strom zostrojiť, stačilo by nám vedieť veľkosti jednotlivých pravidelných stromov, ktoré obsahuje.

**Lemma 3.1.1.** *Majme pravidelný strom veľkosti  $n$ . Hodnotou  $T_0(n)$  si označíme počet vlastných párov tohto stromu, v ktorých 2-pakovanie sa nenachádza ani koreň ani žiaden*

z jeho synov. Potom platí, že

$$\begin{aligned} T_0(n) &= 2^{n/2+1} \cdot 3^{n/2-1}, & n \equiv 0 \pmod{2} \\ T_0(n) &= 2^{(n-1)/2+1} \cdot 3^{(n-1)/2}, & n \equiv 1 \pmod{2} \end{aligned} \quad (3.1)$$

*Dôkaz.* Kľúčom k riešeniu je fakt, že vzdialenosť dvoch listov pravidelného stromu, ktoré nie sú synmi koreňa, je rovná 4. To znamená, že všetky takéto listy môžu patriť do 2-pakovania bez toho aby sa navzájom ovplyvňovali. A zároveň sú to jediné vrcholy, ktoré do 2-pakovania môžu patriť, pretože koreň a jeho synovia sú zakázaní z definície hodnoty  $T_0(n)$ .

Pre každý takýto list máme teda 3 možnosti kam ho umiestniť a pre zvyšné vrcholy máme iba 2 možnosti, lebo sa nemôžu nachádzať v 2-pakovaní. Ostáva nám už len vyriešiť, koľko vrcholov má 3 možnosti. Pre nepárne  $n$  máme jeden koreň a potom  $\frac{n-1}{2}$  dvojíc, z ktorých práve jeden vrchol je listom a patrí do tejto skupiny vrcholov. Pre párne  $n$  musíme odrátať koreň a list, ktorý je synom koreňa. Dostaneme teda  $\frac{n}{2} - 1$  vrcholov, ktoré majú 3 možnosti umiestnenia. Z toho vyplýva uvedený vzorec.  $\square$

V lemme 3.1.1 sme mohli vzorec pre hodnotu  $T_0(n)$  vyjadriť aj vo všeobecnom tvare bez potreby rozlišovania parity čísla  $n$ . Takýto vzorec však obsahuje dolné celé časti a je oveľa neprehľadnejší ako vyššie uvedená alternatíva.

**Lemma 3.1.2.** *Majme pravidelný strom veľkosti  $n$ . Hodnotou  $T_1(n)$  si označíme počet vlastných párov tohto stromu, v ktorých 2-pakovaní sa nenachádza koreň tohto stromu. Potom platí, že*

$$\begin{aligned} T_1(n) &= 2^{(n-2)/2+1} \cdot 3^{(n-2)/2-1} \cdot (n+7), & n \equiv 0 \pmod{2} \\ T_1(n) &= 2^{(n-1)/2+1} \cdot 3^{(n-1)/2-1} \cdot \left(\frac{n-1}{2} + 3\right), & n \equiv 1 \pmod{2} \end{aligned} \quad (3.2)$$

*Dôkaz.* Najprv si zavedme pomocné definície. Tak ako v strom s priemerom 6, aj pravidelné stromy si rozdelíme na úrovne. Na úrovni 0 sa nachádza koreň tohto stromu, na úrovni 1 jeho synovia a na úrovni 2 zvyšné vrcholy. Premennou  $k$  si označme počet vrcholov na úrovni 2. Vzdialenosť medzi dvoma vrcholmi na úrovni 1 je dva, z čoho vyplýva, že v 2-pakovaní môže byť najviac jeden vrchol z tejto úrovne. Vzdialenosť medzi vrcholom úrovne 1 a vrcholom úrovne 2 je jedna ak je druhý vrchol synom prvého a tri v opačnom prípade. Vzdialenosť dvoch vrcholoch na úrovni 2 je štyri.

Nech  $n$  je nepárne. Potom sa na úrovni 1 nachádza  $k$  vrcholov, každý z nich má práve jedného syna na úrovni 2. Rozlíšme dve možnosti podľa toho, či sa v 2-pakovaní nachádza vrchol z úrovne 1. Ak sa tam nenachádza, sme v rovnakej situácii ako v lemme 3.1.1 a máme  $2^{k+1} \cdot 3^k$  možných vlastných párov. Hodnota 2 je umocnená na  $k + 1$  za každý vrchol úrovne 1 a tiež za koreň.

V druhej možnosti je v 2-pakovaní práve jeden vrchol úrovne 1. Máme teda  $k$  možností, ktorý to môže byť. Zvyšné vrcholy z úrovne 1, ktorých je  $k - 1$ , majú len dve možnosti kam ich umiestnime. To isté platí aj pre koreň a syna vybraného vrcholu. No a zvyšných  $k - 1$  vrcholov z úrovne 2 má tri možnosti. Dostávame teda  $k \cdot 2^{k+1} \cdot 3^{k-1}$  vlastných párov. Sčítaním dostaneme výsledok  $2^{k+1} \cdot 3^{k-1} \cdot (k + 3)$ . Nakoniec si uvedomme, že hodnotu  $k$  vypočítame ako  $\frac{n-1}{2}$ , pretože všetky vrcholy mimo koreňa sú rozdelené do dvojíc a dosadením nájdeme požadovaný výraz.

Ostáva možnosť, že  $n$  je párne. V tom prípade máme na úrovni 1 až  $k + 1$  vrcholov. Jeden z nich je totiž list a nemá syna na úrovni 2. Tentokrát musíme rozlíšiť tri prípady – v 2-pakovaní sa nenachádza vrchol z úrovne 1, v 2-pakovaní je vrchol bez syna na úrovni 2, v 2-pakovaní je niektorý iný vrchol z úrovne 1.

V prvom prípade máme  $2^{k+2} \cdot 3^k$  možností, v druhom prípade  $2^{k+1} \cdot 3^k$  a v treťom prípade  $k \cdot 2^{k+2} \cdot 3^{k-1}$ . Všetky tieto hodnoty vypočítame rovnako ako predtým, akurát počítame s vrcholom navyše. Keď sčítame všetky možnosti dokopy, dostaneme  $2^{k+1} \cdot 3^{k-1} \cdot (2k + 9)$ . Takisto vieme, že hodnota  $k$  je pre párne  $n$  rovná  $\frac{n-2}{2}$ , čoho dosadením dostaneme žiadanú rovnosť.  $\square$

Všimnime si, že hodnoty  $T_0(n)$  a  $T_1(n)$  nezahŕňajú disjunktné možnosti, dokonca každý vlastný pár definovaný hodnotou  $T_0(n)$  sa nachádza aj medzi tými určenými  $T_1(n)$ . Túto vlastnosť sme však nepožadovali a ako uvidíme, ani ju potrebovať nebudeme.

**Lemma 3.1.3.** *Majme pravidelný strom veľkosti  $n$ . Hodnotou  $T_2(n)$  si označíme počet vlastných párov tohto stromu, v ktorých 2-pakovaní sa nachádza koreň tohto stromu. Potom platí, že*

$$T_2(n) = 2^{n-1} \tag{3.3}$$

*Dôkaz.* Z definície pravidelných stromov vieme, že hĺbka nášho stromu je najviac 2. To znamená, že každý z vrcholov tohto stromu je od koreňa vzdialený najviac 2. Ak teda v 2-pakovaní leží koreň, nemôže v ňom ležať už žiadny ďalší vrchol. Pre každý

zvyšný vrchol máme teda dve možnosti – buď sa nachádza alebo sa nenachádza v druhej doplnkovej množine. Z toho dostaneme hodnotu  $2^{n-1}$ .  $\square$

Vyššie uvedené lemy nám pomôžu pri dokazovaní nasledovnej, pomerne dôležitej vety.

**Veta 3.1.4.** *Postupnosť  $n_1, n_2 \dots n_k$  je postupnosť kladných čísel. Graf  $G$  je strom, ktorý má ku koreňu pripojených  $k$  podstromov.  $i$ -ty z týchto podstromov je pravidelný strom veľkosti  $n_i$ . Potom počet vlastných párov stromu  $G$  je*

$$\prod_{i=1}^k T_0(n_i) + 2 \prod_{i=1}^k T_1(n_i) + 2 \sum_{i=1}^k \left( T_2(n_i) \cdot \prod_{\substack{1 \leq j \leq k \\ i \neq j}} T_1(n_j) \right) \quad (3.4)$$

*Dôkaz.* Rozdelíme si vlastné páry stromu  $G$  na tri disjunktné množiny. V prvej sú tie, ktoré majú v 2-pakovaní koreň tohto stromu, v druhej tie, ktoré majú v 2-pakovaní práve jedného syna koreňa a v tretej všetky zvyšné, teda tie, ktoré nemajú v 2-pakovaní ani koreň ani žiadneho jeho syna.

Ak je v 2-pakovaní koreň grafu  $G$ , nemôžeme do tohto 2-pakovania vložiť žiaden z vrcholov úrovne 1 a 2. Počet vlastných párov pravidelného stromu, ktoré spĺňajú tieto podmienky nám určuje lemma 3.1.1. Jednotlivé pravidelné podstromy grafu  $G$  navyše môžeme kombinovať. Jediným problémom by totiž bolo, ak by sme do 2-pakovania dali dva korene týchto podstromov. To nám však nehrozí z definície hodnoty  $T_0(n)$ . Počet takýchto vlastných párov je preto  $\prod_{i=1}^k T_0(n_i)$ .

Ako ďalší rozoberme prípad, keď v 2-pakovaní nie je ani koreň ani žiaden jeho syn. To znamená, že pre pravidelný strom veľkosti  $n_i$  hľadáme tie vlastné páry, v ktorých 2-pakovaní nie je ich koreň. To nám udáva hodnota  $T_1(n_i)$  definovaná lemmou 3.1.2. A opäť môžeme takéto vlastné páry dvoch rôznych pravidelných podstromov medzi sebou kombinovať, čím dostaneme  $2 \prod_{i=1}^k T_1(n_i)$  vlastných párov. Nemôžeme totiž zabudnúť prinásobiť dve možnosti, ktoré môže nadobúdať koreň.

Na záver nám ostal prípad, keď je v 2-pakovaní práve jeden syn koreňa. Najskôr si musíme určiť, ktorý to bude, pre všeobecnosť koreň  $i$ -teho podstromu, a potom sčítať možnosti pre všetky hodnoty  $i$ . Pravidelný strom veľkosti  $n_i$  má podľa lemy 3.1.3  $T_2(n_i)$  vlastných párov, v ktorých 2-pakovaní je jeho koreň.



Keď sme do 2-pakovania vybrali jeden z vrcholov na úrovni 1, žiadny ďalší z tejto úrovne tam už nemôže byť, lebo výsledná množina by nebola 2-pakovaním. Na vrcholy na úrovni 2 a 3, ktoré patria do iného pravidelného podstromu však nemáme žiadne obmedzenia. Pre  $j$ -ty podstrom treba preto prinásobiť hodnotu  $T_1(n_j)$ , čo sú práve vlastné páry, v ktorých 2-pakovanie neleží koreň tohto podstromu. Nezabúdajúc na koreň stromu  $G$ , je počet vlastných párov, ktoré majú v 2-pakovaní koreň  $i$ -teho podstromu  $2 \cdot T_2(n_i) \cdot \prod_{\substack{1 \leq j \leq k \\ i \neq j}} T_1(n_j)$ . Sčítaním cez všetky  $i$  dostaneme hľadanú hodnotu.  $\square$

## 3.2 Dolný odhad počtu vlastných párov

V predchádzajúcej časti sme si ukázali ako vyzerajú stromy s priemerom 6, ktoré majú najväčší počet vlastných párov a takisto ako pre takéto typy stromov vypočítať presnú hodnotu tohto počtu. Nevieme však nič o tom, ako veľké majú byť pravidelné podstromy, aby sme tento počet maximalizovali. V tejto časti sa preto pokúsime nájsť triedu stromov, ktoré nám budú slúžiť ako dolný odhad tohto počtu.

### 3.2.1 Experimentálne hľadanie stromov

Napriek tomu, že sme si tvar našich stromov výrazne obmedzili, je ich počet stále pomerne veľký. Stále totiž musíme vyskúšať všetky rozloženia čísla  $n - 1$  na sčítance. A to isté sme robili aj v predošlej kapitole, akurát sme skúšali trochu iné podstromy. A podarilo sa nám nájsť prvých 70 stromov s čo najväčším počtom vlastných párov.

Uvedomme si však, že už v tomto predchádzajúcom generovaní sme dostali iba stromy s priemerom 6. A takmer výlučne sa v nich vyskytovali podstromy veľkostí 2, 3, 5, 7 a 9. A keď sa pozrieme na obrázok 2.2 tieto podstromy boli pravidelné stromy príslušných veľkostí.

Pokúsme sa teda stromy, ktoré budeme generovať ešte viac obmedziť. Presnejšie, budeme generovať iba také stromy s priemerom 6, ktorých pravidelné podstromy majú veľkosti iba 2, 3, 5, 7 a 9. Výhodou tohto prístupu je, že stromov, ktoré spĺňajú tieto podmienky a majú 70 vrcholov je iba 1992. Budeme mať preto možnosť vygenerovať oveľa väčšie stromy a viac nazrieť do štruktúry, ktorú to vytvára.

Takýto algoritmus sme vskutku naprogramovali a spustili pre všetky  $n \leq 250$ . Výsledky tohto testovania si môžete prezrieť v tabuľke 3.1 a 3.2. Výsledky sme rozdelili podľa zvyšku, ktorý dáva číslo  $n$  po delení 7. Z dát sa totiž dá vidieť, že pre všetky väčšie stromy sa pravidelný podstrom veľkosti 7 vyskytuje najčastejšie. Dokonca, ak

je  $n = 7k + 1$ , tak nájdený strom s najväčším počtom vlastných párov obsahuje iba podstromy veľkosti 7.

$n$	veľkosti podstromov	$n$	veľkosti podstromov	$n$	veľkosti podstromov	$n$	veľkosti podstromov
70	(2,2),(7,8),(9,1)	71	(2,1),(5,1),(7,9)	72	(2,2),(7,7),(9,2)	73	(2,1),(7,10)
77	(2,2),(7,9),(9,1)	78	(2,1),(5,1),(7,10)	79	(2,2),(7,8),(9,2)	80	(2,1),(7,11)
84	(2,2),(7,10),(9,1)	85	(2,1),(5,1),(7,11)	86	(5,3),(7,10)	87	(2,1),(7,12)
91	(2,2),(7,11),(9,1)	92	(7,13)	93	(5,3),(7,11)	94	(2,1),(7,13)
98	(2,1),(7,11),(9,2)	99	(7,14)	100	(5,3),(7,12)	101	(2,1),(7,14)
105	(2,1),(7,12),(9,2)	106	(7,15)	107	(5,3),(7,13)	108	(2,1),(7,15)
112	(2,1),(7,13),(9,2)	113	(7,16)	114	(5,3),(7,14)	115	(2,1),(7,16)
119	(2,1),(7,14),(9,2)	120	(7,17)	121	(5,3),(7,15)	122	(7,16),(9,1)
126	(2,1),(7,15),(9,2)	127	(7,18)	128	(5,3),(7,16)	129	(7,17),(9,1)
133	(7,15),(9,3)	134	(7,19)	135	(5,3),(7,17)	136	(7,18),(9,1)
140	(7,16),(9,3)	141	(7,20)	142	(7,15),(9,4)	143	(7,19),(9,1)
147	(7,17),(9,3)	148	(7,21)	149	(7,16),(9,4)	150	(7,20),(9,1)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
238	(7,30),(9,3)	239	(7,34)	240	(7,29),(9,4)	241	(7,33),(9,1)
245	(7,31),(9,3)	246	(7,35)	247	(7,30),(9,4)	248	(7,34),(9,1)

Tabuľka 3.1: Rozloženie stromov s priemerom 6 na podstromy. Dvojica v zátvorke označuje veľkosť podstromu a počet výskytov v rozložení.

### 3.2.2 Počet vlastných párov pre stromy s rovnakými podstromami

Z dát, ktoré sa nám podarilo vygenerovať sme mohli vypožorovať, že najviac sa oplatia stromy, ktorých všetky podstromy sú rovnaké pravidelné stromy, v našom prípade pravidelné stromy s veľkosťou 7. Potrebujeme však vedieť, koľko vlastných párov má takáto trieda stromov pre všeobecnú veľkosť a následne tento počet asymptoticky odhadnúť, aby sme ho vedeli porovnať s predchádzajúcimi výsledkami.

$n$	veľkosti podstromov	$n$	veľkosti podstromov	$n$	veľkosti podstromov
874	(2,1),(5,3),(7,8)	75	(2,2),(7,10)	76	(2,1),(5,2),(7,9)
81	(5,2),(7,10)	82	(2,2),(7,11)	83	(2,1),(5,2),(7,10)
88	(5,2),(7,11)	89	(2,1),(7,11),(9,1)	90	(5,1),(7,12)
95	(5,2),(7,12)	96	(2,1),(7,12),(9,1)	97	(5,1),(7,13)
102	(5,2),(7,13)	103	(2,1),(7,13),(9,1)	104	(5,1),(7,14)
109	(5,2),(7,14)	110	(2,1),(7,14),(9,1)	111	(5,1),(7,15)
116	(5,2),(7,15)	117	(2,1),(7,15),(9,1)	118	(5,1),(7,16)
123	(5,2),(7,16)	124	(7,15),(9,2)	125	(5,1),(7,17)
130	(5,2),(7,17)	131	(7,16),(9,2)	132	(5,1),(7,18)
⋮	⋮	⋮	⋮	⋮	⋮
165	(5,2),(7,22)	166	(7,21),(9,2)	167	(5,1),(7,23)
172	(7,18),(9,5)	173	(7,22),(9,2)	174	(5,1),(7,24)
179	(7,19),(9,5)	180	(7,23),(9,2)	181	(5,1),(7,25)
186	(7,20),(9,5)	187	(7,24),(9,2)	188	(5,1),(7,26)
193	(7,21),(9,5)	194	(7,25),(9,2)	195	(5,1),(7,27)
200	(7,22),(9,5)	201	(7,26),(9,2)	202	(5,1),(7,28)
207	(7,23),(9,5)	208	(7,27),(9,2)	209	(5,1),(7,29)
214	(7,24),(9,5)	215	(7,28),(9,2)	216	(7,23),(9,6)
⋮	⋮	⋮	⋮	⋮	⋮
249	(7,29),(9,5)	250	(7,33),(9,2)	251	(7,28),(9,6)

Tabuľka 3.2: Rozloženie stromov s priemerom 6 na podstromy. Dvojica v zátvorke označuje veľkosť podstromu a počet výskytov v rozložení.

**Definícia 3.2.1.** Strom  $G$  nazveme **strom typu  $x$** , ak má  $kx + 1$  vrcholov a je tvorený koreňom, ku ktorému je pripojených  $k$  pravidelných stromov veľkosti  $x$ .

**Veta 3.2.1.** Nech  $G$  je strom typu  $x$  s  $n$  vrcholmi. Potom počet vlastných párov stromu  $G$  je

$$T_0(x)^{(n-1)/x} + 2 \cdot T_1(x)^{(n-1)/x} + 2 \cdot \frac{n-1}{x} \cdot T_2(x) \cdot T_1(x)^{(n-1)/x-1} \quad (3.5)$$

*Dôkaz.* Strom typu  $x$  s  $n$  vrcholmi je tvorený  $\frac{n-1}{x}$  pravidelnými podstromami veľkosti  $x$ . Dosadením týchto hodnôt do vzorca z vety 3.1.4 dostaneme hľadanú rovnicu.  $\square$

Nás by teraz zaujímalo, aký počet vlastných párov majú stromy typu 7. Potrebujeme preto vyrátať hodnoty  $T_0(7)$ ,  $T_1(7)$  a  $T_2(7)$ . Použitím vzorcov z liem 3.1.1, 3.1.2 a 3.1.3 dostaneme

$$T_0(7) = 432$$

$$T_1(7) = 864$$

$$T_2(7) = 64$$

Majme strom typu 7, ktorý má  $n = 7k + 1$  vrcholov. Potom podľa vety 3.2.1 má nasledovný počet vlastných párov

$$432^k + 2 \cdot 864^k + 128 \cdot k \cdot 864^{k-1}$$

Nás by ale zaujímalo, či sa táto hodnota dá vyjadriť ako  $c^n$ . Vďaka tomu, by sme totiž vedeli asymptoticky odhadnúť rast tejto funkcie. Dostávame nasledovnú limitu, ktorú vypočítame

$$\lim_{k \rightarrow \infty} (432^k + 2 \cdot 864^k + 128 \cdot k \cdot 864^{k-1})^{\frac{1}{7k+1}} = 2^{\frac{5}{7}} 3^{\frac{3}{7}} \approx 2.6272534 \quad (3.6)$$

Trieda stromov typu 7 má preto  $\Theta(2.6272^n)$  vlastných párov. To nám dáva dolný odhad na všeobecné riešenie, ktorý je lepší ako najlepší dosiaľ nájdený odhad  $\Omega(2.6117^n)$ , ktorý sme si ukázali v predchádzajúcej kapitole.

Pre porovnanie nás však zaujíma aj to, ako vyzerá počet vlastných párov pre stromy iných typov. Napríklad zaujímavý môže byť výsledok pre stromy typu 1, keďže to sú grafy známe ako hviezdy, poprípade stromy typu 2, čo sú istým spôsobom tiež hviezdy, ale neporušujú podmienku, že žiadne dva listy nemajú spoločného suseda. V tabuľke 3.3 sa nachádzajú asymptotické odhady pre prvých 15 typov stromov.

Po nazretí do tabuľky 3.3 zistíme, že stromy typu 7 vskutočnosti nie sú najlepším dolným odhadom, aký vieme nájsť. Stromy typu 9 majú vlastných párov ešte o niečo viac. Z tohto zistenia môžeme vyvodiť nasledovnú vetu.

**Veta 3.2.2.** *Hodnota  $pp(n)$  je zdola ohraničená  $\Omega(\sqrt[9]{6048}^n)$ .*

$n$	rýchlosť rastu počtu vlastných párov na stromoch typu $n$
1	2.0
2	2.44948974278
3	2.51984209979
4	2.5755095769
5	2.6051710847
6	2.60430852359
7	2.62725340284
8	2.61099908058
9	2.63135284056
10	2.61033511929
11	2.62903221178
12	2.60686304872
13	2.62424585217
14	2.60231389888
15	2.61857187854

Tabuľka 3.3: Asymptotický rast počtu vlastných párov na stromoch typu  $n$ .

*Dôkaz.* Nech  $n$  je tvaru  $9k + 1$ . Pozrieme sa na počet vlastných párov stromov typu 9, ktoré majú  $n$  vrcholov pre dostatočne veľké  $n$ . Podľa lemy 3.1.1, 3.1.2 a 3.1.3 vieme, že  $T_0(9) = 2592$ ,  $T_1(9) = 6048$  a  $T_2(9) = 256$ . Dosadením týchto hodnôt do vety 3.2.1 zistíme, že počet vlastných párov takéhoto stromu je

$$2592^k + 2 \cdot 6048^k + 2 \cdot 256 \cdot k \cdot 6048^{k-1} \quad (3.7)$$

Nás však zaujíma vyjadrenie tohto počtu v tvare  $c^n$ , kde  $c$  je konštanta. Na zistenie tejto konštanty potrebujeme vyriešiť nasledovnú limitu

$$\lim_{k \rightarrow \infty} (2592^k + 2 \cdot 6048^k + 2 \cdot 256 \cdot k \cdot 6048^{k-1})^{\frac{1}{9k+1}} \quad (3.8)$$

Túto limitu vyriešime tak, že ju odhadneme zdola aj zhora. Pri odhadovaní zhora môžeme všetky konštanty v limite 3.8 nahradiť hodnotou 6048. Vnútri limity preto dostaneme výraz  $(k+3) \cdot 6048^k$ , ktorý musíme odmocniť na  $\frac{1}{9k}$ . Menovateľ tohto zlomku sme si zámerne zmenšili, aby sa nám s ním lepšie pracovalo a horný odhad nám to nepokazí.

Naviac vieme, že  $\lim_{k \rightarrow \infty} k^{\frac{1}{k}} = 1$ , čím dostaneme horné ohraničenie limity 3.8 rovné  $\sqrt[9]{6048}$ .

Pri odhade zdola môžeme z tela limity vyškrtnúť všetko okrem člena  $6048^k$ , čím dostaneme limitu  $\lim_{k \rightarrow \infty} 6048^{\frac{k}{9k+1}}$ . Člen v exponente prenásobíme výrazom  $\frac{1/k}{1/k}$ , čím dostaneme zlomok  $\frac{1}{9+\frac{1}{k}}$ . Hodnota  $\frac{1}{k}$  však ide v limite k 0, preto dostávame dolné ohraničenie  $\sqrt[9]{6048}$ .

Keďže sme limitu 3.8 ohraničili zdola aj zhora rovnakým číslom, toto číslo je jej riešenie. Počet vlastných párov stromov typu 9 je preto asymptoticky rovný  $\Theta(\sqrt[9]{6048}^n)$ . A keďže stromy typu 9 je existujúca trieda grafov, je táto hodnota dolným ohraničením hodnoty  $pp(n)$ .  $\square$

### 3.2.3 Porovnanie stromov typu 7 a stromov typu 9

Ako sme videli v tabuľke 3.3 pri stromoch malého typu mali najlepší asymptotický počet vlastných párov stromy typu 9. Napriek tomu nám v experimentoch vychádzali stromy typu 7, ako môžeme vidieť v tabuľke 3.1 a 3.2. Preto by mohlo byť zaujímavé sa pozrieť na to, prečo to tak bolo a aký je vlastne rozdiel v počte vlastných párov týchto dvoch stromov.

Nech  $n = 7 \cdot 9 \cdot k + 1$ . Potom existuje aj strom typu 7 aj strom typu 9, ktorý ma takýto počet vrcholov. Vyjadrime si teraz počet vlastných párov oboch týchto stromov podľa vety 3.2.1. Pre strom typu 7 to bude

$$pp_7(k) = 432^{9k} + 2 \cdot 864^{9k} + 128 \cdot 9 \cdot k \cdot 864^{9k-1} \quad (3.9)$$

a pre strom typu 9

$$pp_9(k) = 2592^{7k} + 2 \cdot 6048^{7k} + 2 \cdot 256 \cdot 7 \cdot k \cdot 6048^{7k-1} \quad (3.10)$$

Otázkou teda je, pre aké prirodzené  $k$  platí  $pp_9(k) - pp_7(k) \geq 0$ . Rozborom funkcie  $pp_9(k) - pp_7(k)$  ľahko nahliadneme, že to platí pre všetky  $k$  väčšie ako 6. Aby sa teda stromy typu 9 prejavili ako stromy s väčším počtom vlastných párov, potrebujeme pracovať so stromami, ktoré majú rádovo 400 vrcholov. Naše experimenty však pracovali so stromami, ktoré mali iba 250 vrcholov, preto sa v nich vyskytovali prevažne podstromy veľkosti 7.

### 3.3 Nahrádzanie pravidelných podstromov

Na konci predchádzajúcej časti sme si ukázali, že pre dostatočne veľké stromy sa oplatia stromy typu 9 oproti stromom typu 7. A podľa tabuľky 3.3 sme videli, že to platí dokonca o stromoch typu 11. Otázkou však je, či to platí aj pre stromy väčších typov.

Ešte zaujímavejšia je však otázka, ako vyzerá  $n$  vrcholový strom s priemerom 6, ktorý má najväčší počet vlastných párov. Nemôže byť trieda stromov typu 9 nie len dolným odhadom, ale asymptoticky presným?

Ak by sme chceli dokázať takúto vlastnosť pre stromy s priemerom 6, musíme dokázať, že nie je výhodné, aby takýto strom obsahoval pravidelný podstrom inej veľkosti ako 9. Poprípade, že takýchto stromov obsahuje iba konštantne veľa. Toto budeme dokazovať tak, že ukážeme, že nahradenie pravidelného podstromu inými menšími pravidelnými podstromami nezmenší počet vlastných párov.

#### 3.3.1 Všeobecný počet vlastných párov

Dôkazy, ktoré budeme robiť, budú lokálne meniť časť stromu. Potrebujeme však vedieť vyrátať, aký je počet vlastných párov v tomto strome bez ohľadu na to, ako vyzerá časť stromu, ktorú nemeníme. Na to nám budú slúžiť nasledujúce tri hodnoty.

**Definícia 3.3.1.** *Nech  $T$  je ľubovoľný zakorenený strom. Hodnotou  $M_0(T)$  si označíme počet vlastných párov stromu  $T$ , v ktorých 2-pakovaní sa nachádza koreň tohto stromu.*

**Definícia 3.3.2.** *Nech  $T$  je ľubovoľný zakorenený strom. Hodnotou  $M_1(T)$  si označíme počet vlastných párov stromu  $T$ , v ktorých 2-pakovaní sa nachádza práve jeden syn koreňa tohto stromu.*

**Definícia 3.3.3.** *Nech  $T$  je ľubovoľný zakorenený strom. Hodnotou  $M_2(T)$  si označíme počet vlastných párov stromu  $T$ , v ktorých 2-pakovaní sa nenachádza ani koreň tohto stromu, ani žiaden syn koreňa.*

Ako si iste všimnete, tieto hodnoty sú veľmi podobné tomu, čo sme definovali pri návrhu dynamického programovania alebo v lemmach 3.1.1, 3.1.2 a 3.1.3. Napriek tomu sú tieto hodnoty špecifické, lebo vypovedajú o ľubovoľnom strome  $T$ .

Vďaka týmto trom hodnotám teraz vieme vysloviť nasledujúcu lemmu.

**Lemma 3.3.1.** *Nech  $T$  je ľubovoľný strom s koreňom  $v$ . Ku koreňu  $v$  pripojíme pravidelné podstromy veľkostí  $n_1, n_2 \dots n_k$  a novovzniknutý strom označíme  $T'$ . Potom platí, že*

$$\begin{aligned} pp(T') &= M_0(T) \cdot \prod_{i=1}^k T_0(n_i) \\ &+ M_1(T) \cdot \prod_{i=1}^k T_1(n_i) \\ &+ M_2(T) \cdot \left( \prod_{i=1}^k T_1(n_i) + \sum_{i=1}^k \left( T_2(n_i) \cdot \prod_{1 \leq j \leq k, i \neq j} T_1(n_j) \right) \right) \end{aligned} \quad (3.11)$$

*Dôkaz.* Budeme postupovať podobne ako v dôkaze vety 3.1.4. Všetky vlastné páry si rozdelíme na 3 skupiny – v 2-pakovaní sa nachádza koreň  $v$ , v 2-pakovaní patriacom  $T$  sa nachádza práve jeden syn koreňa a v 2-pakovaní patriacom  $T$  sa nenachádza ani koreň ani jeho syn z  $T$ . Tieto tri možnosti totiž zodpovedajú definícii hodnôt  $M_0(T)$ ,  $M_1(T)$  a  $M_2(T)$ .

Pre každú možnosť nám už len zostáva prinásobiť možnosti za pridané podstromy. Ak sa v 2-pakovaní nachádza koreň, pravidelný podstrom nemôže do 2-pakovania prispieť ani koreňom ani synom. Počet takýchto vlastných párov nám udáva hodnota  $T_0(n_i)$ .

Ak sa v 2-pakovaní už nachádza nejaký syn koreňa, pravidelné podstromy nemôžu do 2-pakovania prispieť svojím koreňom. To znamená, že máme  $T_1(n_i)$  možností pre  $i$ -ty podstrom a tieto možnosti musíme medzi sebou prenásobiť.

No a ak sa v 2-pakovaní stromu  $T$  nenachádza ani koreň ani žiaden jeho syn, tak máme pre naše podstromy dve možnosti. V jednom z nich nepridáme do 2-pakovania žiadneho syna koreňa v  $T'$ , na čo opäť použijeme hodnoty  $T_1(n_i)$ . V druhej možnosti si vyberieme niektorého syna, toho vložíme do 2-pakovania, pričom on nám pridá  $T_2(n_i)$  možnosti a ostatné nemôžu pridať do 2-pakovania svoj koreň, čo bude hodnota  $T_1(n_j)$ .

Po sčítaní všetkých týchto hodnôt dostaneme požadovaný výraz.  $\square$

Na záver tejto časti si dokážeme ešte jednu lemmu, ktorá bude veľmi užitočná v nasledujúcich dôkazoch.

**Lemma 3.3.2.** *Nech  $T$  je ľubovoľný strom. Potom platí, že  $2 \cdot M_0(T) \leq M_2(T)$ .*

*Dôkaz.* Nech  $(S, X)$  je vlastný pár patriaci do množiny  $M_0(T)$ . To znamená, že koreň stromu  $T$ , ktorý si označíme  $v$ , leží v 2-pakovaní  $S$ . Z čoho vyplýva, že v  $S$  neleží žiaden



syn koreňa. Z tohto vlastného páru vieme vytvoriť dva iné vlastné páry,  $(S - \{v\}, X)$  a  $(S - \{v\}, X \cup \{v\})$ .

Oba tieto vlastné páry neobsahujú v svojom 2-pakovaní ani koreň  $v$ , ani žiadneho jeho syna. To znamená, že sú to vlastné páry započítané do množiny  $M_2(T)$ . Z každého vlastného páru  $(S, X)$  patriacemu  $M_0(T)$  vieme preto vytvoriť dva nové vlastné páry patriace do  $M_2(T)$ . Navyše, všetky takto vytvorené vlastné páry sú navzájom rôzne. Preto platí zadaná nerovnosť.  $\square$

### 3.3.2 Nahrádzanie veľkých podstromov

Ako prvé by sme chceli ohraničiť veľkosť najväčšieho pravidelného podstromu, ktorý sa môže nachádzať v strome, ktorý má najväčší počet vlastných párov. Zoberieme si preto strom  $T$ , ktorý bude tvorený stromom  $T'$ , ktorému pod koreň zavesíme pravidelný strom veľkosti  $x$ . Vďaka hodnotám  $M_0(T')$ ,  $M_1(T')$  a  $M_2(T')$  budeme vedieť porovnať, čo sa stane s počtom vlastných párov stromu  $T$  ak vrcholy tohto pravidelného podstromu preusporiadame do inej štruktúry.

Využitie tejto myšlienky si ukážeme na ďalej veľmi dôležitej lemme.

**Lemma 3.3.3.** *Majme ľubovoľný strom  $G$  s koreňom  $v$ . Strom  $G_1$  dostaneme z  $G$  tak, že ku koreňu  $v$  pripojíme pravidelný podstrom veľkosti  $x$ . Strom  $G_2$  dostaneme z  $G$  tak, že ku koreňu  $v$  pripojíme pravidelný podstrom veľkosti  $9$  a  $x - 9$ .*

*Potom platí, že ak je  $x \geq 18$ , tak má strom  $G_2$  aspoň toľko vlastných párov ako strom  $G_1$ .*

*Dôkaz.* Podľa lemy 3.3.1 vieme vyjadriť hodnoty  $pp(G_1)$  a  $pp(G_2)$ . To čo potrebujeme ukázať je, že pre  $x \geq 18$  platí nerovnosť  $pp(G_2) - pp(G_1) \geq 0$ .

Dôkaz si v tomto momente rozdelíme na dve časti podľa parity čísla  $x$ . Nech  $x$  je nepárne. Potom potrebujeme vyjadriť hodnotu  $pp(G_2) - pp(G_1)$ . Dosadením do vety 3.3.1 dostaneme

$$\begin{aligned}
pp(G_2) - pp(G_1) = & M_0(G) \cdot \left( -2^{(x-1)/2+1} \cdot 3^{(x-1)/2-1} \right) \\
& + M_1(G) \cdot \left( 2^{(x-1)/2+1} \cdot 3^{(x-5)/2-1} \cdot \left( \frac{5x+9}{2} - 41 \right) \right) \\
& + M_2(G) \cdot \left( 2^{(x-1)/2+1} \cdot 3^{(x-5)/2-1} \cdot \left( \frac{5x+9}{2} - 41 \right) \right) \\
& + 2^3 \cdot 2^{(x-1)/2+1} \cdot 3^{(x-11)/2-1} \cdot (x-2) \\
& + 2^{x-5} \cdot 3^3 \cdot 7 - 2^{x-1} \Big) \quad (3.12)
\end{aligned}$$

V tejto rovnici ľahko nahliadneme, že koeficient pri člene  $M_0(G)$  je záporný pre ľubovoľné  $x$ . Koeficient pri člene  $M_1(G)$  je kladný ak je kladný výraz  $\frac{5x+9}{2} - 41$ , čo platí pre všetky  $x \geq 15$ . V koeficiente pri člene  $M_2(G)$  máme niekoľko sčítancov. Prvý z nich je rovnaký ako pri člene  $M_1(G)$  a preto je kladný taktiež pre všetky  $x \geq 15$ . Druhý z nich je zjavne kladný pre všetky  $x \geq 3$ . No a tretí člen je rozdiel dvoch hodnôt, ale keďže  $3^3 \cdot 7 > 2^4$ , tento člen je kladný pre všetky  $x$ .

Aby sme ukázali, že výraz  $pp(G_2) - pp(G_1)$  je kladný pre dostatočne veľké  $x$ , musíme ukázať, že kladný príspevok  $M_1(G)$  a  $M_2(G)$  je väčší ako záporný  $M_0(G)$ . K tomu použijeme tvrdenie lemy 3.3.2 a nahradíme hodnotu  $M_2(G)$  hodnotou  $2M_0(G)$ , ktorá určite nie je väčšia. Takisto zanedbáme všetko okrem druhého člena koeficientu pri  $M_2(G)$ , z čoho vyplynie, že potrebujeme dokázať nerovnicu

$$M_0(G) \cdot \left( 2^{(x-1)/2+1} \cdot 3^{(x-1)/2-1} \right) \leq 2M_0(G) \cdot \left( 2^3 \cdot 2^{(x-1)/2+1} \cdot 3^{(x-11)/2-1} (x-2) \right) \quad (3.13)$$

ktorú upravíme na jednoduchší tvar  $3^5 \leq 2^4 \cdot (x-2)$  Táto nerovnica zjavne platí, pre  $x \geq 18$ .

Pre nepárne  $x$  väčšie ako 18 preto platí, že strom  $G_2$  má aspoň toľko vlastných párov ako strom  $G_1$ .

Ešte potrebujeme vyriešiť prípad, keď je  $x$  párne. Dosadením do vety 3.3.1 dostaneme nasledovnú rovnicu

$$\begin{aligned}
pp(G_2) - pp(G_1) &= M_0(G) \cdot 0 \\
&+ M_1(G) \cdot \left( 2^{(x-2)/2+1} \cdot 3^{(x-4)/2+1} \cdot (4x - 49) \right) \\
&+ M_2(G) \cdot \left( 2^{(x-2)/2+1} \cdot 3^{(x-4)/2+1} \cdot (4x - 49) \right. \\
&\quad \left. + 2^3 \cdot 2^{x/2+1} \cdot 3^{(x-10)/2-1} \left( \frac{x-10}{2} + 3 \right) \right. \\
&\quad \left. + 2^{x-5} \cdot 3^3 \cdot 7 - 2^{x-1} \right)
\end{aligned} \tag{3.14}$$

V tomto prípade jasne vidíme, že koeficient pri člene  $M_1(G)$  je kladný pre  $x \geq 13$  a pre takto veľké  $x$  sú kladné aj všetky sčítance v koeficiente pre  $M_2(G)$ . Tým pádom, pre párne  $x$  väčšie ako 13 má strom  $G_2$  aspoň toľko vlastných párov ako strom  $G_1$ .  $\square$

**Veta 3.3.4.** *Pre každé prirodzené číslo  $n$  existuje strom  $G$  s najväčším počtom vlastných párov, ktorého žiaden podstrom nie je pravidelný strom veľkosti väčšej ako 18.*

*Dôkaz.* Zoberme si ľubovoľný strom  $G$ , ktorý má najväčší počet vlastných párov. Ak je niektorý z jeho podstromov pravidelný strom väčší ako 18, použitím lemy 3.3.3 ho rozdelíme na dva menšie podstromy bez toho, aby sme zmenšili počet vlastných párov tohto stromu. Tento postup opakujeme, kým sa v ňom ďalej nenachádzajú ďalšie priveľké podstromy.  $\square$

Veta 3.3.4 má pomerne veľký význam pri stromoch priemeru 6, keďže výrazne limituje ich vzhľad. Takisto má silné implikácie aj do všeobecného prípadu. Uvedomme si navyše, že odhad 18 je pravdepodobne zbytočne premrštený, pretože pri dôkaze sme veľa vecí zanedbávali. Ako si však ukážeme v ďalšej časti, stačí nám odhad, ktorý je konštantný a dostatočne malý, čo číslo 18 spĺňa.

### 3.3.3 Nahrádzanie malých podstromov

V predchádzajúcej časti sme si ukázali pomerne silnú vetu 3.3.4, ktorá nám zaručila, že v strome s najväčším počtom vlastných párov sa nemôže nachádzať pravidelný podstrom väčší ako 18. Zatiaľ ale nevieme nič povedať o pravidelných podstromoch menšej veľkosti.

Ako vieme, strom s priemerom 6 je tvorený koreňom, pod ktorým sú zavesené pravidelné stromy rôznych veľkostí. Môže sa však stať, že riešenie maximalizujúce počet vlastných párov má polovicu týchto podstromov veľkosti 9 a druhú polovicu veľkosti 7. Popríklad pomedzi to premiešané nejaké iné malé podstromy.

My sa však snažíme ukázať, že najlepšie riešenie je použiť čo najviac podstromov veľkosti 9, presnejšie, že len konštantne veľa podstromov stromu s priemerom 6 je inej veľkosti. Chceli by sme preto pre každý pravidelný podstrom s menej ako 18 vrcholmi vedieť, koľko najviac takýchto podstromov sa môže nachádzať v optimálnom riešení bez toho, aby sme ich nevedeli rozložiť na menšie podstromy, ktoré neznížia počet vlastných párov.

Predstavme si, že máme  $k$  pravidelných podstromov veľkosti  $x$ . Označme si multimnožinu obsahujúcu  $k$  krát hodnotu  $x$  ako  $A$ . Podľa lemy 3.3.1 vieme zistiť, aký počet vlastných párov má takýto strom. Označme si túto hodnotu  $p(A) = a_0 \cdot M_0(G) + a_1 \cdot M_1(G) + a_2 \cdot M_2(G)$ . Následne prerozdeliť všetkých  $kx$  vrcholov iným spôsobom, čím dostaneme inú multimnožinu pravidelných podstromov, ktorú si označme  $B$ . Z lemy 3.3.1 vieme opäť vypočítať počet vlastných párov, ak je pod koreň stromu  $G$  zavesená nie množina  $A$ , ale množina  $B$ . Túto hodnotu označíme  $p(B) = b_0 \cdot M_0(G) + b_1 \cdot M_1(G) + b_2 \cdot M_2(G)$ .

Použijeme lemmu 3.3.2 a pozrime sa na rozdiel  $p(B) - p(A)$ . Dostaneme výraz  $M_1(G) \cdot (b_1 - a_1) + M_0(G) \cdot (b_0 + 2b_2 - a_0 - 2a_2)$ . Ak je tento výraz väčší alebo rovný 0, tak sa nám oplatí vymeniť množinu podstromov  $A$  za množinu podstromov  $B$ , keďže sa nezníži počet vlastných párov. A hodnota  $k$  bude predstavovať hornú hranicu počtu podstromov veľkosti  $x$ , ktoré sa môžu vyskytovať v našom strome s priemerom 6.

Toto všetko však boli dobre popísateľné algoritmické kroky. Potrebujeme predsa len vypočítať koeficienty  $a_0$ ,  $a_1$ ,  $a_2$ ,  $b_0$ ,  $b_1$  a  $b_2$  a potom overiť, či  $b_1 - a_1 \geq 0$  a  $b_0 + 2b_2 - a_0 - 2a_2 \geq 0$ . Dokonca program bude vedieť ľahko vygenerovať všetky možné preusporiadania vrcholov množiny  $A$  a skúsiť všetky prípustné množiny  $B$ .

Takýto algoritmus sme skutočne implementovali a jeho výsledok sme zhrnuli do tabuľky 3.4. Vidíme, že pre každú veľkosť pravidelného podstromu menšiu ako 18 máme pomerne malý odhad najväčšieho počtu takto veľkých podstromov, ktoré sa môžu nachádzať v našich stromoch.



*Dôkaz.* Z dôkazu vety 3.2.2 vieme, že stromy typu 9 s  $n$  vrcholmi majú  $\Theta(\sqrt[9]{6048}^n)$  vlastných párov. Z vety 3.3.4 a tabuľky 3.4 vieme, že každý strom priemeru 6, ktorý má najväčší počet vlastných párov obsahuje iba konštantne veľa pravidelných podstromov, ktoré nie sú veľkosti 9.

Zoberme si strom  $G$  s priemerom 6, ktorý má na  $n$  vrcholoch najväčší počet vlastných párov. Ako vieme, tento strom je tvorený niekoľkými pravidelnými podstromami. Iba konštantne veľa z nich je inej veľkosti ako 9. Odstráňme tieto podstromy z grafu  $G$ , čím získame strom typu 9, ktorý si označíme  $G'$ . Vieme, že  $G'$  má  $n - c$  vrcholov, pre nejakú konštantu  $c$ . Teda počet vlastných párov stromu  $G'$  je  $\Theta(\sqrt[9]{6048}^{n-c})$ . Hodnota  $\sqrt[9]{6048}^{n-c}$  sa však od  $\sqrt[9]{6048}^n$  líši iba konštantných násobkom, preto je počet vlastných párov stromu  $G'$ , a teda aj  $G$ ,  $\Theta(\sqrt[9]{6048}^n)$ . Túto hodnotu vieme zhruba vyčíslieť na  $\Theta(2.63135^n)$ .  $\square$

### 3.4.1 Algoritmus na nájdenie stromu priemeru 6 s najväčším počtom vlastných párov

Vyššie uvedené poznatky však majú ešte jeden zaujímavý dôsledok. Nielenže vieme povedať asymptoticky, aký najväčší počet vlastných párov má strom s  $n$  vrcholmi, ale vieme vypočítať aj to, ako tento strom vyzerá a koľko presne tých vlastných párov má.

Ako vieme, každý strom priemeru 6, ktorý má najväčší počet vlastných párov má iba konštantne veľa podstromov inej veľkosti ako 9. Naskytá sa nám preto možnosť vyskúšať všetky tieto možnosti a pre každú z nich presne spočítať, koľko vlastných párov má takto vytvorený strom. Ako vieme z tabuľky 3.4, každý strom priemeru 6 s najväčším počtom vlastných párov, má najviac jeden podstrom veľkosti 1, dva podstromy veľkosti 2, dva podstromy veľkosti 3, dva podstromy veľkosti 4, päť podstromov veľkosti 5, jeden podstrom veľkosti 6, päťdesiattri podstromov veľkosti 7, jeden podstrom veľkosti 8, jeden podstrom veľkosti 10, štyri podstromy veľkosti 11 a jeden podstrom veľkosti 13. To nám dáva iba 1 399 680 možností, ktoré musíme prezrieť.

Na to, aby sme mohli na počítanie počtu vlastných párov použiť lemmu 3.3.1, potrebujeme vedieť vypočítať hodnoty  $M_0(G)$ ,  $M_1(G)$  a  $M_2(G)$ , kde  $G$  je strom typu 9, pod ktorý zavesíme podstromy, ktoré nie sú veľké 9.

**Lemma 3.4.2.** *Nech  $G$  je strom typu 9, ktorý má pod koreň zavesených  $k$  pravidelných podstromov veľkosti 9. Potom*

$$\begin{aligned}M_0(G) &= 2^{5k} \cdot 3^{4k} \\M_1(G) &= k \cdot 2^{5k+4} \cdot 3^{3k-3} \cdot 7^{k-1} \\M_2(G) &= 2^{5k} \cdot 3^{3k} \cdot 7^k\end{aligned}\tag{3.15}$$

*Dôkaz.* Postupujeme podľa definície hodnôt  $M_0(G)$ ,  $M_1(G)$  a  $M_2(G)$ , pričom využívame hodnoty z liem 3.1.1, 3.1.2 a 3.1.3. Dosadením dostaneme príslušné odvodenia.  $\square$

Vďaka tejto lemme vieme navrhnúť algoritmus, ktorý postupne prechádza všetky možnosti, ako vyzerá časť podstromu, ktorá neobsahuje pravidelné podstromy veľkosti 9 a nájde tú, ktorá maximalizuje počet vlastných párov. Počas celého algoritmu potrebujeme umocniť iba konštantne veľa prvkov na hodnotu  $n$ . Ak preto budeme predpokladať, že aritmetické operácie vykonáme v konštantnom čase a zanedbáme zložitosť výstupu, ktorého veľkosť závisí na tom, ako si ho definujeme, dostaneme algoritmus s časovou zložitosťou  $\Theta(\log n)$ .

# Kapitola 4

## Implikácie pre všeobecný prípad

V predchádzajúcej kapitole sme sa zamerali na stromy s priemerom 6. K tomuto kroku nás navádzali všetky experimenty, ktoré sme spravili, keďže v žiadnom z nich sme nenašli strom s väčším priemerom. Ako sme však videli pri stromoch typu 7 a 9, niektoré vlastnosti sa začnú prejavovať až pri takých počtoch vrcholov, ktoré nie sme schopní efektívne spracovávať.

Zaujímalo nás preto, či sa z výsledkov, ktoré sme odvodili pre stromy priemeru 6 nedajú vyvodiť tvrdenia aj pre všeobecné stromy. Minimálne tvrdenia o nahrádzaní podstromov boli aplikovateľné aj vo všeobecnom prípade ako nám ukazuje veta 3.3.4 a tabuľka 3.4. Samozrejme najlepšie by bolo, ak by sa nám podarilo dokázať, že stromy typu 9 majú najväčší počet vlastných párov aj vo všeobecnosti.

Ako však nahliadneme v tejto kapitole, takýto predpoklad nie je správny. Ukážeme si preto, ako sa dajú postupy a výsledky z 3. kapitoly aplikovať na nájdenie ešte lepšieho dolného odhadu.

### 4.1 Limitné hľadanie hodnoty $c$ pre stromy s rovnakými podstromami

Ako sme videli na grafoch s priemerom 6, asymptoticky presný počet vlastných párov udávali stromy typu 9, ktoré vznikli tak, že sme pod koreň zavesili iba pravidelné podstromy veľkosti 9. Naviac, zdá sa intuitívne, že ak obmedzíme veľkosti podstromov zavesených pod koreňom, pre dostatočne veľké grafy sa oplatí opakovať práve jeden konkrétny podstrom a že sa pre tento podstrom budú dať ukázať podobné nahradzovacie lemmy, ako sme dokázali pre pravidelné stromy veľkosti 9.



Síce takéto tvrdenie nevieme dokázať, zdá sa dostatočne intuitívne a potvrdené príkladom zo stromov priemeru 6, aby sme sa ním skúsili zaoberať. Naviac, ako uvidíme, výrazne nám to zjednoduší počítanie. V celej tejto kapitole sa budeme preto zaoberať iba takými stromami, ktoré majú pod koreňom zavesené rovnaké podstromy.

**Definícia 4.1.1.** *Nech  $G$  je zakorenený strom. **Strom typu  $G$**  nazveme každý strom, ktorý je tvorený jedným koreňom, pod ktorým je za koreň zavesených niekoľko stromov  $G$ .*

Pred tým, ako vyslovíme nasledujúcu lemmu, rozšírme si definície hodnôt  $T_0(n)$ ,  $T_1(n)$  a  $T_2(n)$  z liem 3.1.1, 3.1.2 a 3.1.3. Hodnota  $T_0(G)$  označuje počet vlastných párov stromu  $G$ , v ktorých 2-pakovaní sa nenachádza koreň stromu  $G$ , ani žiaden jeho syn. Hodnota  $T_1(G)$  označuje počet vlastných párov stromu  $G$ , v ktorých 2-pakovaní sa nenachádza koreň stromu  $G$ . No a hodnota  $T_2(G)$  označuje počet vlastných párov stromu  $G$ , ktoré v 2-pakovaní obsahujú koreň stromu  $G$

**Lemma 4.1.1.** *Majme strom typu  $G$ , ktorý obsahuje  $k$  podstromov  $G$ . Počet vlastných párov takéhoto stromu je*

$$T_0(G)^k + 2 \cdot T_1(G)^k + 2 \cdot k \cdot T_2(G) \cdot T_1(G)^{k-1} \quad (4.1)$$

*Dôkaz.* Vlastné páry nášho stromu rozdelíme na tri množiny – tie čo v 2-pakovaní obsahujú koreň tohto stromu, tie čo v 2-pakovaní nemajú ani koreň ani žiadneho jeho syna a tie, ktoré majú v 2-pakovaní práve jedného syna koreňa. Veľkosti týchto množín potom vypočítame s použitím hodnôt  $T_0(G)$ ,  $T_1(G)$  a  $T_2(G)$  priamočiario.  $\square$

**Lemma 4.1.2.** *Pre ľubovoľný strom  $G$  platí:*

$$\begin{aligned} T_0(G) &\leq T_1(G) \\ 2T_2(G) &\leq T_1(G) \end{aligned} \quad (4.2)$$

*Dôkaz.* Množina  $T_0(G)$  obsahuje všetky vlastné páry stromu  $G$ , ktoré v 2-pakovaní nemajú ani koreň stromu  $G$ , ani žiadneho jeho syna. Každý takýto vlastný pár však priamo z definície patrí aj do množiny  $T_1(G)$ .

V množine  $T_2(G)$  sú všetky vlastné páry, ktoré majú v 2-pakovaní koreň stromu  $G$ . Z každého takéhoto vlastného páru  $(S, X)$  však vieme vyrobiť dva vlastné páry  $(S - \{v\}, X)$  a  $(S - \{v\}, X \cup \{v\})$ , kde  $v$  je koreň  $G$ . Tieto nové vlastné páry patria do množiny  $T_1(G)$ , z čoho vyplýva žiadaná nerovnosť.  $\square$

Obe práve dokázané lemmy sa nám hodia pri vyslovení a dokázaní nasledujúcej, dôležitej vety. Ako uvidíme, je to vlastne zovšeobecnenie vety 3.2.2, ktorú sme dokázali pre stromy typu 9.

**Veta 4.1.3.** *Nech  $G$  je zakorenený  $k$  vrcholový graf. Počet vlastných párov stromov typu  $G$  je potom  $\Theta(\sqrt[k]{T_1(G)}^n)$ .*

*Dôkaz.* Označme si  $n$  vrcholový strom typu  $G$ , ktorý obsahuje  $x$  podstromov  $G$  ako  $P_x$ . Zaujímá nás taká hodnota  $c$ , že  $c^n$  je počet vlastných párov grafu  $P_x$ . A keďže chceme zistiť jej asymptotický odhad, budeme túto hodnotu počítat' limitne pre zväčšujúce sa  $n$ . A keďže  $n = xk + 1$ , cieľom bude vyrátať limitu

$$\lim_{x \rightarrow \infty} pp(P_x)^{1/(xk+1)} \quad (4.3)$$

Hodnotu  $pp(P_x)$  vieme vyjadriť pomocou lemmy 4.1.1. Dosadením dostaneme

$$\lim_{x \rightarrow \infty} (T_0(G)^x + 2 \cdot T_1(G)^x + 2 \cdot x \cdot T_2(G) \cdot T_1(G)^{x-1})^{1/(xk+1)} \quad (4.4)$$

Pokúsme sa odhadnúť limitu 4.4 zhora. Výsledná hodnota sa určite nezmenší, ak namiesto  $xk + 1$  budeme odmocňovať hodnotu  $xk$ . No a podľa lemmy 4.1.2 môžeme hodnotu  $T_0(G)$  aj  $2T_2(G)$  nahradiť hodnotou  $T_1(G)$ . Dosadením dostaneme

$$\lim_{x \rightarrow \infty} ((x + 3) \cdot T_1(G)^x)^{1/(xk)} = T_1(G)^{1/k} \quad (4.5)$$

Poslednú rovnosť dostaneme, keď si uvedomíme, že výraz  $(x + 3)^{1/x}$  sa blíži limitne k hodnote 1.

Limitu 4.4 musíme ešte odhadnúť zdola. Vďaka tomu môžeme zanedbať všetko okrem stredného člena  $T_1(G)^x$ . Exponent limity následne pre násobíme výrazom  $\frac{(1/x)}{(1/x)}$ , čím dostaneme

$$\lim_{x \rightarrow \infty} T_1(G)^{1/(k+1/x)} = T_1(G)^{1/k} \quad (4.6)$$

Posledná rovnosť vyplýva z toho, že výraz  $1/x$  sa limitne blíži k 0.

Limitu 4.4 sme teda ohraničili zhora aj zdola rovnakou hodnotou  $T_1(G)^{1/k}$ . Z toho vyplýva, že počet vlastných párov stromu typu  $G$  je naozaj  $\Theta(\sqrt[k]{T_1(G)}^n)$ .  $\square$

## 4.2 Stromy maximalizujúce $T_1(G)$

Z vety 4.1.3 vyplýva dôležité pozorovanie. Ak chceme nájsť strom typu  $G$ , ktorý má čo najväčší počet vlastných párov, musíme maximalizovať hodnotu  $T_1(G)^{1/k}$ , kde  $k$  je počet vrcholov stromu  $G$ . A ak si zafixujeme hodnotu  $k$ , hľadáme taký strom  $G$  s  $k$  vrcholmi, ktorý maximalizuje hodnotu  $T_1(G)$ .

Tak ako v 2. kapitole by sme teda mohli vygenerovať všetky stromy s daným počtom vrcholov a spočítať pre ne, hodnotu  $T_1(G)$ . V tomto prípade však potrebujeme generovať všetky zakorenené stromy, keďže hodnota  $T_1(G)$  sa líši od zvoleného koreňa aj pre inak izomorfné stromy. Rozšíriť naše generovanie o túto podmienku však nie je problém.

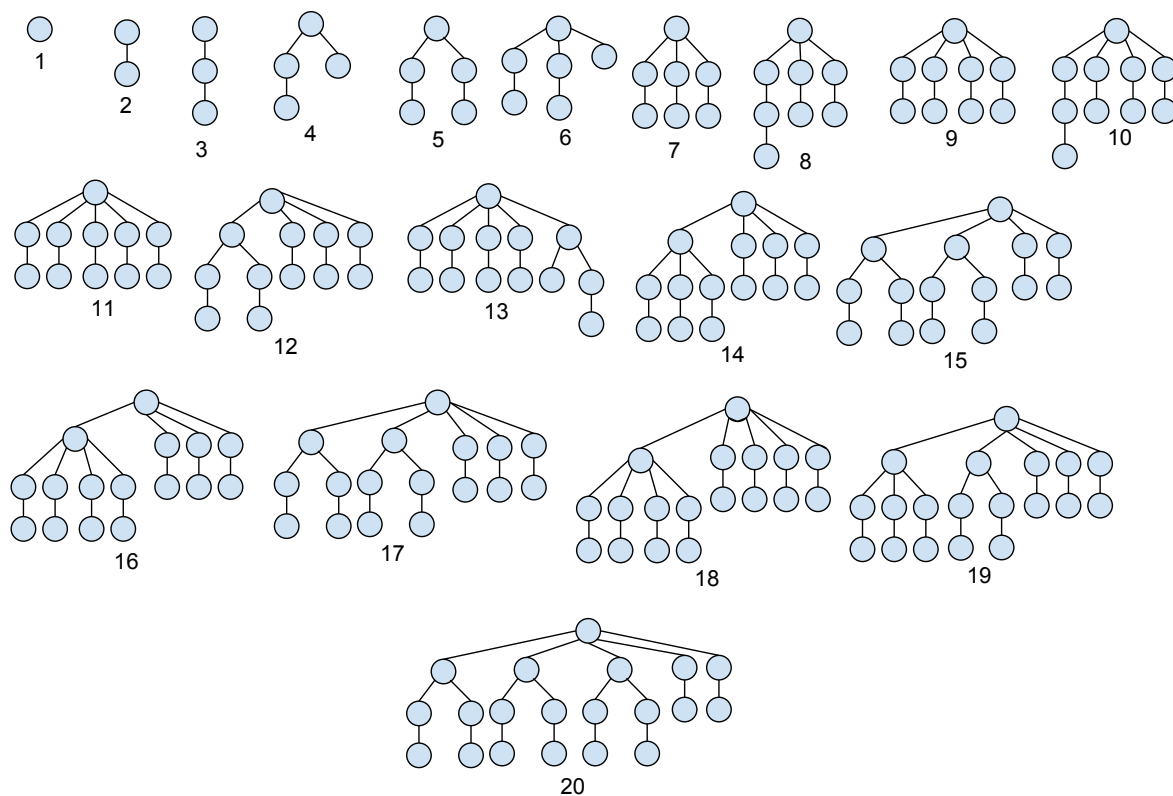
Čo sa týka samotného výpočtu hodnoty  $T_1(G)$ , zistíme, že vieme opäť použiť to isté dynamické programovanie. V ňom sme počítali tri hodnoty.  $P(v, 0)$  bol počet vlastných párov podstromu vrchola  $v$ , ktoré mali v 2-pakovaní vrchol  $v$ .  $P(v, 1)$  bol počet vlastných párov podstromu vrchola  $v$ , ktoré v 2-pakovaní obsahovali jedného zo synov vrchola  $v$ . No a  $P(v, 2)$  bol počet vlastných párov, ktoré nemali v 2-pakovaní ani vrchol  $v$  ani žiadneho jeho syna. Je teda jasné, že ak má strom  $G$  koreň vo vrchole  $v$ , tak  $T_1(G) = P(v, 1) + P(v, 2)$ . Taktiež si všimnime, že  $T_2(G) = P(v, 0)$ .

Naprogramovali sme preto algoritmus, ktorý generoval všetky stromy s určitým počtom vrcholov a potom hľadal ten, ktorý mal najväčšiu hodnotu  $T_1(G)$ . Tento algoritmus sa nám podarilo spustiť pre počty vrcholov najviac 20. Na obrázku 4.1 si môžete pozrieť ako vyzerajú stromy s najväčším  $T_1(G)$  pre jednotlivé počty vrcholov. V tabuľke 4.1 si potom môžete prezrieť, aká je hodnota  $T_1(G)^{1/k}$  pre tieto jednotlivé stromy. Vidíme, že táto hodnota je najväčšia pre 19 vrcholový strom  $G_{19}$ . Počet vlastných párov stromu typu  $G_{19}$  je preto  $\Theta(2.63507^n)$ .

Ako si môžeme všimnúť porovnaním obrázkov 2.2 a 4.1, grafy, ktoré maximalizujú  $T_1(G)$  sa podobajú, ak priamo nerovnajú stromom s najväčším počtom vlastných párov. Toto zistenie nie je až tak prekvapivé, keď si uvedomíme, že platí nasledovné:

$$pp(G) = P(v, 0) + P(v, 1) + P(v, 2) = T_2(G) + T_1(G) \leq \frac{3}{2}T_1(G) \quad (4.7)$$

Posledná nerovnica vyplýva z 4.1.2. Je teda jasné, že stromy s najväčším počtom

Obr. 4.1: Stromy s najväčšou hodnotou  $T_1(G)$  pre daným počet vrcholov.

$k$	$T_1(G)^{1/k}$	$k$	$T_1(G)^{1/k}$
1	2.0	11	2.62903
2	2.44948	12	2.63214
3	2.51984	13	2.62917
4	2.57550	14	2.63408
5	2.60517	15	2.63107
6	2.60430	16	2.63469
7	2.62725	17	2.63355
8	2.61818	18	2.63398
9	2.63135	19	2.63507
10	2.62529	20	2.63334

Tabuľka 4.1: Hodnota  $T_1(G)^{1/k}$  pre  $k$  vrcholový graf s najväčším  $T_1(G)$ .

vlastných párov majú aj pomerne veľa vlastných párov, ktoré v 2-pakovaní neobsahujú koreň stromu.

V predchádzajúcich experimentoch sa nám však podarilo vygenerovať pomerne veľké množstvo stromov, ktoré mali veľa vlastných párov. Najviac aspoň vzhľadom na danú

štruktúru. Či už to bolo 70 rekurzívnych stromov alebo 250 stromov s pravidelnými podstromami. Rozhodli sme sa preto vyskúšať aký počet vlastných párov majú stromy typu  $G$ , ak  $G$  je jeden z týchto predtým vygenerovaných stromov.

Ako najlepší sa ukázal 127 vrcholový strom, ktorý bol tvorený koreňom, pod ktorým bolo zavesených 18 pravidelných stromov veľkosti 7. Stromy, ktorých všetky podstromy sú takto vyzerajúce stromy majú  $\Theta(2.639165^n)$  vlastných párov.

**Veta 4.2.1.** *Hodnota  $pp(n)$  je zdola ohraničená  $\Omega(2.639165^n)$ .*

*Dôkaz.* Nech  $G$  je strom typu 7 obsahujúci 18 pravidelných podstromov veľkosti 7. Strom typu  $G$  má potom  $\Theta(2.639165^n)$  vlastných párov. To môžeme vypočítať pomocou vety 4.1.3 a hodnoty  $T_1(G)$  pre tento konkrétny graf.  $\square$

# Záver

V našej práci sme sa zaoberali problémom hľadania stromov s maximálnym počtom vlastných párov. Za pomoci experimentov na stromoch s malým počtom vrcholov sa nám podarilo nájsť triedu stromov, ktorá nám poskytla lepší dolný odhad na počet vlastných párov  $\Omega(2.63916^n)$ . Toto vylepšenie je výrazným posunutím oproti predošlému odhadu  $\Omega(2.6117^n)$ . Taktiež sme zistili, že trieda stromov s priemerom 6 má vlastných párov  $\Theta(6048^{n/9})$ .

Síce sme nezlepšili horný odhad algoritmu Junosza-Szaniawski a kolektívu, priblížili sme mu odhad dolný. V prípade, že by sa vyskytol iný algoritmus riešiaci problém  $L(2, 1)$ -farbenia, bude preň ľahšie dokázať, že je lepší ako nami študovaný algoritmus. Takisto sme bližšie preskúmali samotnú štruktúru problému počtu vlastných párov a prišli sme s niekoľkými novými postupmi, ktoré by sa dali použiť v nadväzujúcich prácach.

Ako sme videli v 4. kapitole, ako dôležitý parameter zakoreneného stromu sa ukázala hodnota  $T_1(G)$ . V budúcnosti by sa preto oplatilo bližšie pozrieť na stromy maximalizujúce túto hodnotu, poprípade na stromy typu  $G$  s obmedzenou veľkosťou podstromu  $G$ . No a v neposlednej rade, by sa hodnota  $T_1(G)$  mohla využiť na dokázanie lepšieho horného odhadu, keďže už v dôkaze prezentovanom Junosza-Szaniawskim a kolektívom sa namiesto hodnoty  $T_1(n)$  používa určite väčšia hodnota  $pp(n)$ .

# Literatúra

- [1] Gunnar Brinkmann, Kris Coolsaet, Jan Goedgebeur, and Hadrien Mélot. House of graphs: A database of interesting graphs. *Discrete Applied Mathematics*, 161(1-2):311–314, 2013.
- [2] Jiří Fiala, Petr A. Golovach, and Jan Kratochvíl. *Automata, Languages and Programming: 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005. Proceedings*, chapter Distance Constrained Labelings of Graphs of Bounded Treewidth, pages 360–372. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [3] Jerrold R. Griggs and Roger K. Yeh. Labelling graphs with a condition at distance 2. *SIAM Journal on Discrete Mathematics*, 5(4):586–595, 1992.
- [4] W. K. Hale. Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68(12):1497–1514, Dec 1980.
- [5] Toru Hasunuma, Toshimasa Ishii, Hirotaka Ono, and Yushi Uno. *Algorithms - ESA 2009: 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, chapter A Linear Time Algorithm for L(2,1)-Labeling of Trees, pages 35–46. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [6] Frédéric Havet, Martin Klazar, Jan Kratochvíl, Dieter Kratsch, and Mathieu Liedloff. Exact algorithms for l(2,1)-labeling of graphs. *Algorithmica*, 59(2):169–194, 2009.
- [7] Konstanty Junosza-Szaniawski, Jan Kratochvíl, Mathieu Liedloff, Peter Rossmanith, and Pawel Rżazewski. Fast exact algorithm for L(2,1)-labeling of graphs. *Theoretical Computer Science*, 505:42–54, 2013.
- [8] Konstanty Junosza-Szaniawski and Pawel Rżazewski. *Combinatorial Algorithms: 21st International Workshop, IWOCA 2010, London, UK, July 26-28, 2010, Revised Selected Papers*, chapter On Improved Exact Algorithms for L(2,1)-Labeling of Graphs, pages 34–37. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

- [9] Konstanty Junosza-Szaniawski and Pawel Rzazewski. On the complexity of exact algorithm for 1 (2, 1)-labeling of graphs. *Information Processing Letters*, 111(14):697 – 701, 2011.
- [10] Daniel Král'. The channel assignment problem with variable weights. *SIAM Journal on Discrete Mathematics*, 20(3):690–704, 2006.
- [11] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, {II}. *Journal of Symbolic Computation*, 60(0):94 – 112, 2014.
- [12] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356.