

Největší klika v neorientovaném grafu

Lukáš Lev

Fakulta informačních technologií Vysokého učení technického v Brně
Božetěchova 1/2, 612 00 Brno – Královo Pole
256660@vutbr.cz



28. listopadu 2024

Zadání varianty

Vytvořte program pro hledání největší kliky v neorientovaném grafu. Pokud existuje více řešení, **nalezněte všechna**. Výsledky prezentujte vhodným způsobem. Součástí projektu bude **načítání grafů** ze souboru a vhodné **testovací grafy**. V dokumentaci uveďte teoretickou složitost úlohy a **porovnejte** ji s experimentálními výsledky.

Řešení musí obsahovat:

- algoritmus pro nalezení všech největších klik v neorientovaném grafu,
- načítání a zapisování grafů do souboru,
- testovací grafy,
- porovnání teoretické složitosti s experimentálními výsledky.

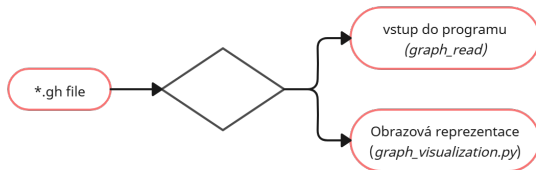
Vybrané algoritmy:

- hrubá síla (*brute force*),
- zpětné vyhledávání (*backtracking*).

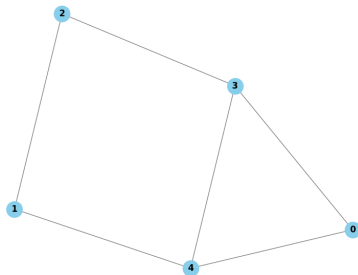
Reprezentace neorientovaného grafu pomocí matice sousednosti (kapitola 4.3)

- výhody
 - jednoduchost
- nevýhody
 - $O(n^2)$ prostorové složitosti
 - charakter úlohy \implies iterace všemi podgrafy $\implies O(2^n)$

$$\mathbf{M}_s = \begin{pmatrix} h_{00} & h_{01} & h_{02} & h_{03} & h_{04} \\ h_{10} & h_{11} & h_{12} & h_{13} & h_{14} \\ h_{20} & h_{21} & h_{22} & h_{23} & h_{24} \\ h_{30} & h_{31} & h_{32} & h_{33} & h_{34} \\ h_{40} & h_{41} & h_{42} & h_{43} & h_{44} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \quad (1)$$

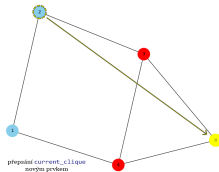
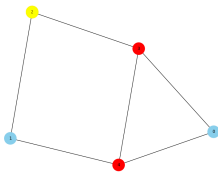
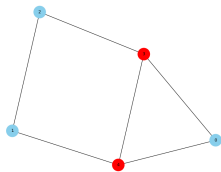


5
0 0 0 1 1
0 0 1 0 1
0 1 0 1 0
1 0 1 0 1
1 1 0 1 0



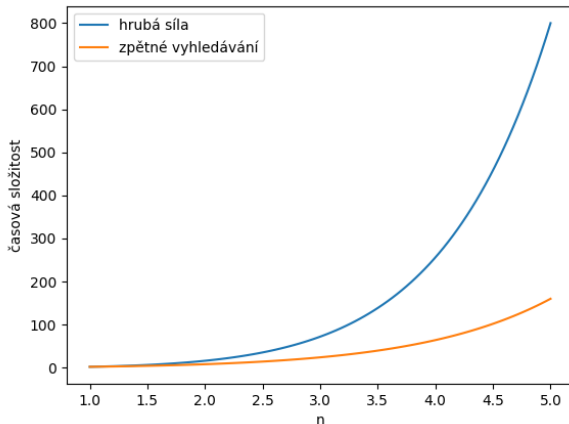
- bitová maska
 - $10_{(10)} = 01010_{(2)} \implies u_0 u_1 u_2 u_3 u_4$
- časová složitost $\mathbf{O(2^n \cdot n^2)}$
 - 2^n možných podgrafů
 - nejhůře kontrola $\binom{n}{2} = \frac{n(n-1)}{2} \implies n^2$ pro podgraf velikosti n
- prostorová složitost $\mathbf{O(2^n \cdot n)}$
 - pro 2^n cyklů uloženo n prvků

- časová složitost $O(2^n \cdot n)$
 - 2^n možných podgrafů
 - nejhůře až n iterací
- prostorová složitost $O(2^n \cdot n)$
 - pro 2^n cyklů uloženo n prvků



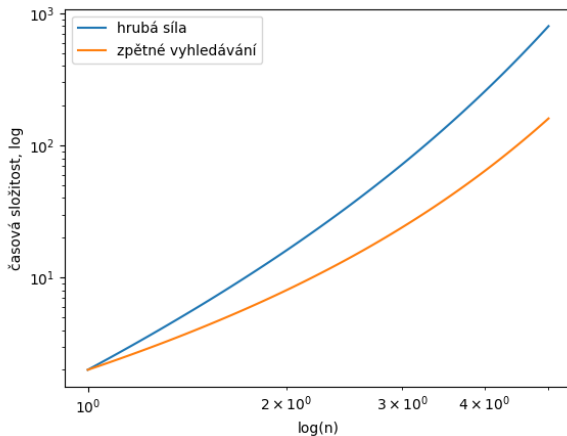
- 1 $\forall n \in \langle 30; 300 \rangle \cap \mathbb{N}, \rho = 0, 5$, algoritmus metody zpětného vyhledávání
- 2 $\forall n \in \langle 5; 30 \rangle \cap \mathbb{N}, \rho = 0, 5$, algoritmus metody hrubé síly
- 3 $\forall n \in \langle 5; 59 \rangle \cap \mathbb{N}, \forall \rho \in \bigcup_{k=1}^9 0, 1 \cdot k$, algoritmus metody zpětného vyhledávání
- 4 $\forall n \in \langle 5; 92 \rangle \cap \mathbb{N}, \forall \rho \in \bigcup_{k=1}^9 0, 1 \cdot k$, algoritmus metody hrubé síly

Teoretický předpoklad (horní hranice stanovena funkcí Omikron)

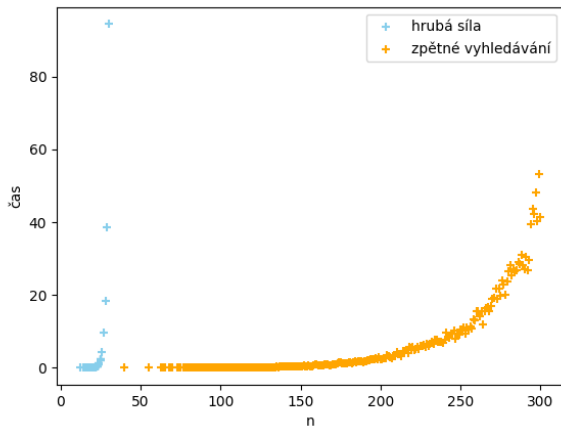


Teoretický předpoklad (funkce Omikron)

logaritmické škálování

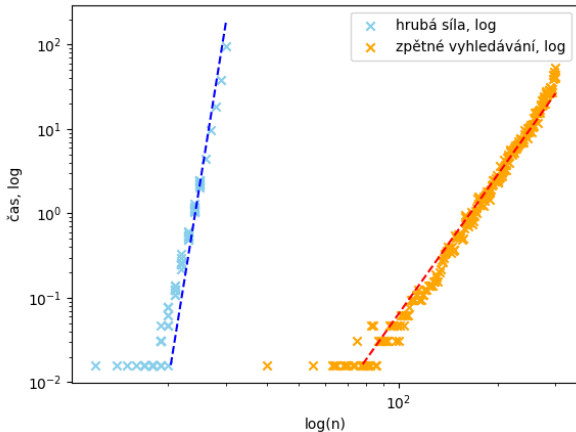


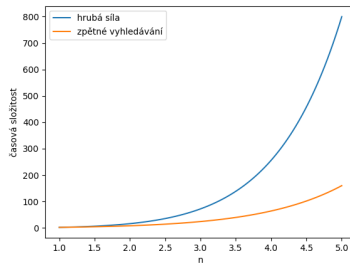
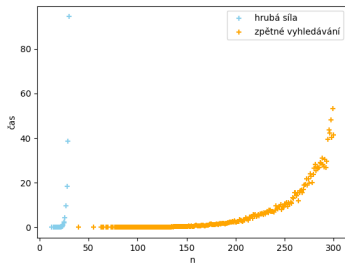
Výsledek experimentu

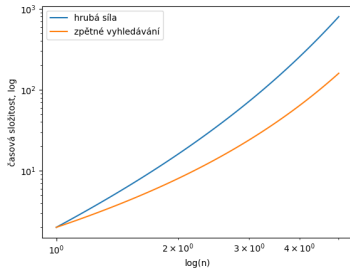
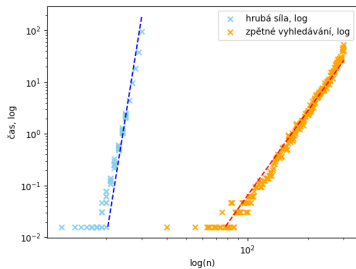


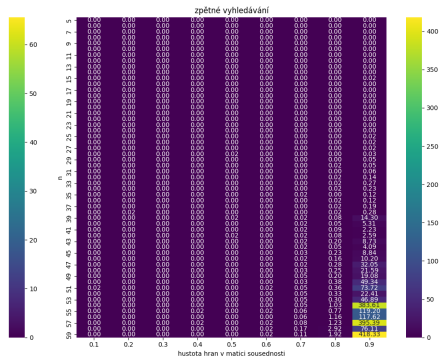
Výsledek experimentu

logaritmické škálování

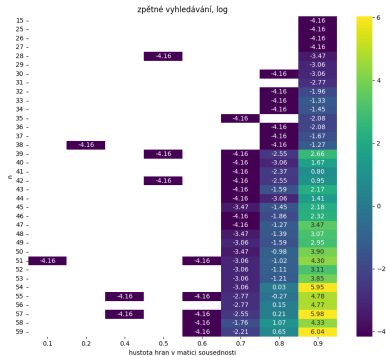
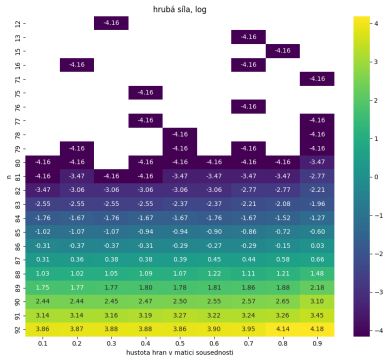








Výsledek experimentu logaritmické škálování



- porovnání teoretických předpokladů časové složitosti s experimentem
- určení vhodné aplikace pro každý z algoritmů
- nastínění dalších kroků pro přesnější porovnání teorie s experimentem

Děkuji za pozornost.