Hamiltonova cesta / cyklus

LUKÁŠ DRAHNÍK – XDRAHN00
JAN VAVŘINA – XVAVRI10

Teorie

- Hamiltonovská **cesta** v grafu ***G*** je cesta, která obsahuje každý vrchol ***V*** v grafu ***G*** právě jednou a žádná z využitých cest **P** se neopakuje
- pokud v ***G*** existuje cesta délky ***V – 1*** splňující první bod, je **cesta** Hamiltonovská
- pokud začíná a končí ve stejném vrcholu a je délky ***V***, jedná se o **cyklus**, kterého délka je **V**
- Hamiltonovská **cesta** je také vždy nejdelší cestou v grafu
- spadá do kategorie ***NP-úplné***

Vylučující podmínky

Počet vrcholů - **> 2**

Stupeň každého z vrcholů pro cestu - **>= 1**

Stupeň každého z vrcholů pro cyklus - **>= 2**

Stupeň vrcholu: počet hran, které jsou s daným vrcholem spojené

Při nesplnění se program nepouští!

Potvrzující podmínky

Diracova - každý vrchol má stupeň alespoň ½ celkového počtu vrchol

Oreho - každá dvojice vrcholů nespojených hranou má součet stupňů alespoň jako je celkový počet vrcholů

Pósova - pro každé přirozené číslo $k < \frac{1}{2}$ celkého počtu vrcholů existuje počet vrcholů, jejichž stupeň nepřevyšuje k, menší než k

Pouze jako informace navíc! *(při zapnutém debug módu)*

Vstupní data a zobrazování

```
A B
A G
B C
B G
B H
C D
C J
D E
D H
E F
E I
E J
F I
G H
G J
H I
I J
```

***.in**

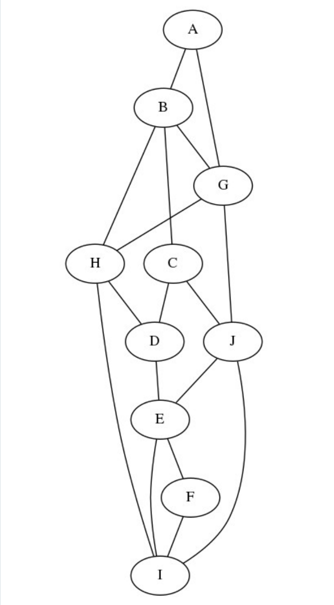
Vstupní data a zobrazování

```
A B
A G
B C
B G
B H
C D
C J
D E
D H
E F
E I
E J
F I
G H
G J
H I
I J
```

→ (python script) →

```
strict graph "" {
        A -- B;
        A -- G;
        B -- G;
        B -- C;
        B -- H;
        G -- H;
        G -- J;
        C -- D;
        C -- J;
        H -- D;
        H -- I;
        D -- E;
        J -- E;
        J -- I;
        E -- F;
        E -- I;
        F -- I;
}
```

→ Graphviz →



***.in** ***.dot** ***.png**

Implementovaný algoritmus

- začátek prohledávání na počátečním vrcholu
- postupný průchod všemi zbylými hranami (rekurzivní zavolání)

- **bez uvedení** počátečního nebo cílového vrcholu => **první vrchol**

Implementovaný algoritmus (vývoj)

Urychlení výpočtu programu:

- využití dostupných vláken – **zavrhnuto** (malý počet výpočetních kroků, příliš velká režie)

Snižování paměťové náročnosti:

- úprava datových typů (využíváním pouze potřebných bitů)

Počet vrcholů	Hrany	Prozkoumané vrcholy	Doba trvání[s]	Počet řešení	Počet alokací	Alokovaná paměť[B]
3	3	5	0,008	2	32	6766
4	6	16	0,009	6	58	8948
5	10	65	0,009	24	126	17478
6	15	326	0,009	120	410	64204
7	21	1957	0,010	720	2068	392246
8	28	13700	0,023	5040	13842	3091836
9	36	109601	0,132	40320	109778	28090078
10	45	986410	1,219	362880	986626	284130788
11	55	9864101	14,063	3628800	9864360	3156573535
12	66	108505112	187,34	39916800	108505418	38193881957

Experimentální ověření složitosti

Testováno na grafech se 3 – 12 vrcholy

Děkujeme za pozornost

Prostor pro dotazy

Zdroje

Hamiltonova cesta / cyklus

LUKÁŠ DRAHNÍK – XDRAHN00
JAN VAVŘINA – XVAVRI10

- Pozdravit - dobrý den
- Představit se - dovolte nám, aby jsme se představili, já jsem a pracoval jsem v týmu společně s kolegou Honzou Vavřinou
- Zadání – zadání našeho skupinového projektu bylo najít Hamiltonovu cestu / potažmo cyklus se zadanými vrcholy v neorientovaném grafu

Teorie

- Hamiltonovská **cesta** v grafu ***G*** je cesta, která obsahuje každý vrchol ***V*** v grafu ***G*** právě jednou a žádná z využitých cest **P** se neopakuje
- pokud v ***G*** existuje cesta délky ***V – 1*** splňující první bod, je **cesta** Hamiltonovská
- pokud začíná a končí ve stejném vrcholu a je délky ***V***, jedná se o **cyklus**, kterého délka je **V**
- Hamiltonovská **cesta** je také vždy nejdelší cestou v grafu
- spadá do kategorie ***NP-úplné***

- Neorientovaný graf – neorientovaný graf znamená, že můžeme při existující hraně z bodu A do bodu B procházet tuto hranu / toto spojení dvou vrcholů / uzlů oběma směry (nikde se neuvádějí šipečky)

- Hamiltonova cesta je propojení všech vrcholů tak, že každý obsáhneme pouze jednou, délka této cesty je poté o 1 menší než počet vrcholů (snadno představitelné na 3 bodech, projdu 2 hrany), každou hranu můžeme použít pouze jednou

- Hamiltonův cyklus je poté specifický případ Hamiltonovy cesty, která začíná a končí ve stejném vrcholu a splňuje zmíněné podmínky, délka cyklz je počet vrcholů

- Kategorie NP-úplný

Vylučující podmínky

Počet vrcholů - **> 2**

Stupeň každého z vrcholů pro cestu - **>= 1**

Stupeň každého z vrcholů pro cyklus - **>= 2**

Stupeň vrcholu: počet hran, které jsou s daným vrcholem spojené

Při nesplnění se program nepouští!

- Vylučující podmínky - snaha eliminovat puštění algoritmu, který může běžet několik dní i déle

Potvrzující podmínky

Diracova - každý vrchol má stupeň alespoň ½ celkového počtu vrchol

Oreho - každá dvojice vrcholů nespojených hranou má součet stupňů alespoň jako je celkový počet vrcholů

Pósova - pro každé přirozené číslo k < ½ celkého počtu vrcholů existuje počet vrcholů, jejichž stupeň nepřevyšuje k, menší než k

Pouze jako informace navíc! *(při zapnutém debug módu)*

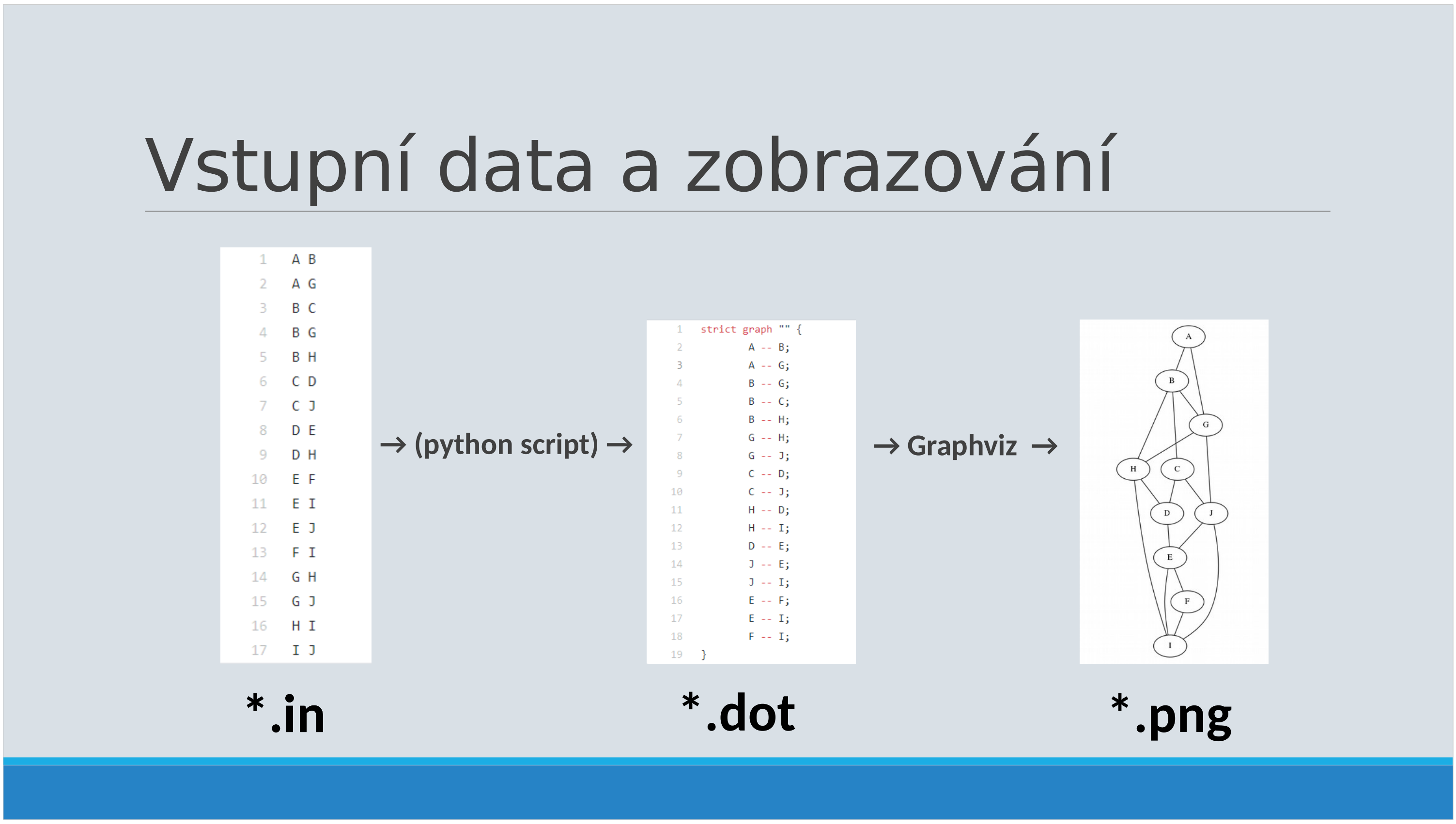
Vstupní data a zobrazování

```
A B
A G
B C
B G
B H
C D
C J
D E
D H
E F
E I
E J
F I
G H
G J
H I
I J
```

***.in**

- Formát vstupního souboru – formát vstupního souboru je následující, co řádek to hrana mezi dvěma vrcholy. Vrchol _ mezera_ vrchol. Koncovka vstupního souboru je .in odvozená od slova input

- Inline zápis - graf je možné taky zapsat inline v terminálu



- Vstupní soubor pro Graphviz – grafy jsou vysualizované pomocí knihovny Graphviz, která je poskytovaná i jako modul pro python a ten byl právě využitý.

- Zmiňovaný vstupní soubor s koncovkou .in byl převeden jednoduchým scriptem v pythonu do formátu dot a ten předán jako vstup pro Graphviz.

.DOT – Graph Description Language

Implementovaný algoritmus

- začátek prohledávání na počátečním vrcholu
- postupný průchod všemi zbylými hranami (rekurzivní zavolání)

- **bez uvedení** počátečního nebo cílového vrcholu => **první vrchol**

Implementovaný algoritmus (vývoj)

Urychlení výpočtu programu:

- využití dostupných vláken – **zavrhnuto** (malý počet výpočetních kroků, příliš velká režie)

Snižování paměťové náročnosti:

- úprava datových typů (využíváním pouze potřebných bitů)

- Využívání dostupných vláken - během vývoje programu proběhl pokus využít dostupná vlákna, ale po krátkém prototypování jsem zjistil, že počet výpočetních kroků před spuštěním dalšího vlákna není takový, aby se vlákno vyplatilo (režie byla příliš velká)

- Snižování paměťové náročnosti - úpravou typů (využívání pouze potřebných bitů)

Počet vrcholů	Hrany	Prozkoumané vrcholy	Doba trvání[s]	Počet řešení	Počet alokací	Alokovaná paměť[B]
3	3	5	0,008	2	32	6766
4	6	16	0,009	6	58	8948
5	10	65	0,009	24	126	17478
6	15	326	0,009	120	410	64204
7	21	1957	0,010	720	2068	392246
8	28	13700	0,023	5040	13842	3091836
9	36	109601	0,132	40320	109778	28090078
10	45	986410	1,219	362880	986626	284130788
11	55	9864101	14,063	3628800	9864360	3156573535
12	66	108505112	187,34	39916800	108505418	38193881957

Experimentální ověření složitosti

Testováno na grafech se 3 – 12 vrcholy

- Použité testovací grafy - pro experimentální data byly použity grafy s propojením každého vrcholu s každým, tedy nejhorší možný případ (od 3 vrcholů do 13)

- Prostorová složitost – lze vycházet z alokované paměti / počtu alokování uvedené zde v tabulce - (ukázat!!)

- Časová složitost – sloupec prozkoumané vrcholy odpovídá počtu zavolání rekurze programu, metodě algorithm() - (ukázat!!)

Děkujeme za pozornost

Prostor pro dotazy

Zdroje