

Vizualizace grafových algoritmů

Tématem práce je vizualizace grafových algoritmů, konkrétně prohledávání do hloubky, do šířky, Dijkstrův algoritmus a hledání Eulerovské kružnice v neorientovaných grafech (nakreslení grafu jedním tahem).

Všechny algoritmy jsou krokovatelné uživatelem, který zadá libovolný graf (viz dále), a poté může krok po kroku sledovat, jak algoritmus probíhá, a případně ho může pustit znovu z jiného počátečního vrcholu.

Kompilace & spuštění

Pro kompilaci je potřeba překladač který umí C++11, např. Clang/LLVM verze 3.5, který je dostupný v labu. Dále je potřeba Qt5, zde by na konkrétní verzi nemělo záležet, ale Qt4 nestačí.

Kompilace a spuštění se pak provede následovně

```
$ qmake . $ make $ ./build/debug/graphite
```

Testováno na různých distribucích Linuxu, včetně labu, a OS X 10.10.

TODO spuštění bez Qt?

Základní ovládání

Program se ovládá klávesnicí i myší, kde všechny příkazy jdou zadat buď přes hlavní menu

SCREENSHOT

a nebo pomocí klávesové zkratky. Zde je jejich úplný seznam:

- Vygenerování náhodného grafu **Ctrl-R G**.
- Náhodná orientace hran **Ctrl-R D**.
- Vygenerování náhodného Eulerovského grafu **Ctrl-R E**.
- Náhodné ohodnocení hran **Ctrl-R W**.
- Odstranění orientace hran **Ctrl-R U**.
- Přidání vrcholu **A**.
- Spojení dvou vrcholů hranou **C**. Je potřeba napřed vybrat první, zmáčknout **C**, vybrat druhý, a zmáčknout **C** znovu.
- Smazání vrcholu nebo hrany **D**. (nejprve je potřeba hranu nebo vrchol vybrat kliknutím myši.)

- Označení počátečního vrcholu S .
- Start/Restart algoritmu R .
- Krok algoritmu N .
- Změna orientace hrany 0 (nejprve je potřeba hranu vybrat kliknutím myši.)
- Nastavení ohodnocení hrany $0-9$, nejprve je ale potřeba mít vybraný Dijkstrův algoritmus, jinak se ohodnocení hran nezobrazí, a poté kliknout na vybrané ohodnocení (*ne na hranu*).

Grafy je také možné uložit do souboru **Ctrl-S** a znovu načíst **Ctrl-S**, přičemž se zachová i rozložení vrcholů v prostoru (pokud je uživatel přesunul.)

Používání

Nejjednodušší je vybrat jeden z příložených grafů v souboru, a otevřít jej přes **File -> Open**, např. kompletní graf na 5 vrcholech v souboru **examples/k5.g**

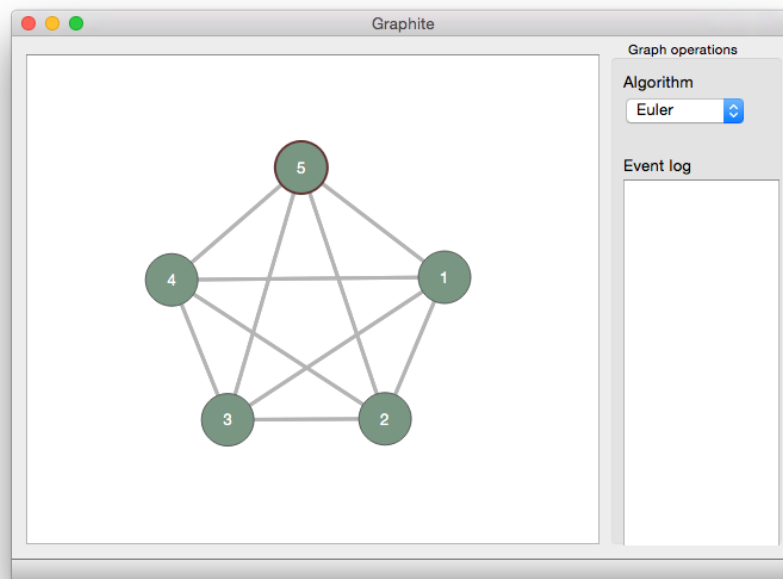


Figure 1: k5-screenshot

v seznamu algoritmů vybrat Eulerovský tah

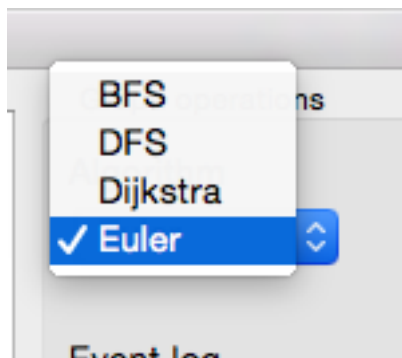


Figure 2: alg-selection

kliknout na libovolný vrchol, vybrat ho jako počáteční (stisknutím F), inicializovat algoritmus (stisknutím R), a poté již krokovat stisknutím N.

V libovolnou chvíli je možné znovu stiknout R, čímž se algoritmus resetuje a začne pracovat odznova. **Pokud uživatel graf jakkoliv změnil v průběhu algoritmu, je nutné algoritmus resetovat stisknutím R.**

TODO - resetovat ručně

TODO - popsat další alg

Generování náhodných grafů

Aby bylo možné aplikaci jednoduše používat, obsahuje možnost vygenerování náhodného souvislého grafu (žádný ze zabudovaných algoritmů nedává smysl vizualizovat na nesouvislých grafech.)

Graf je generován následujícím způsobem:

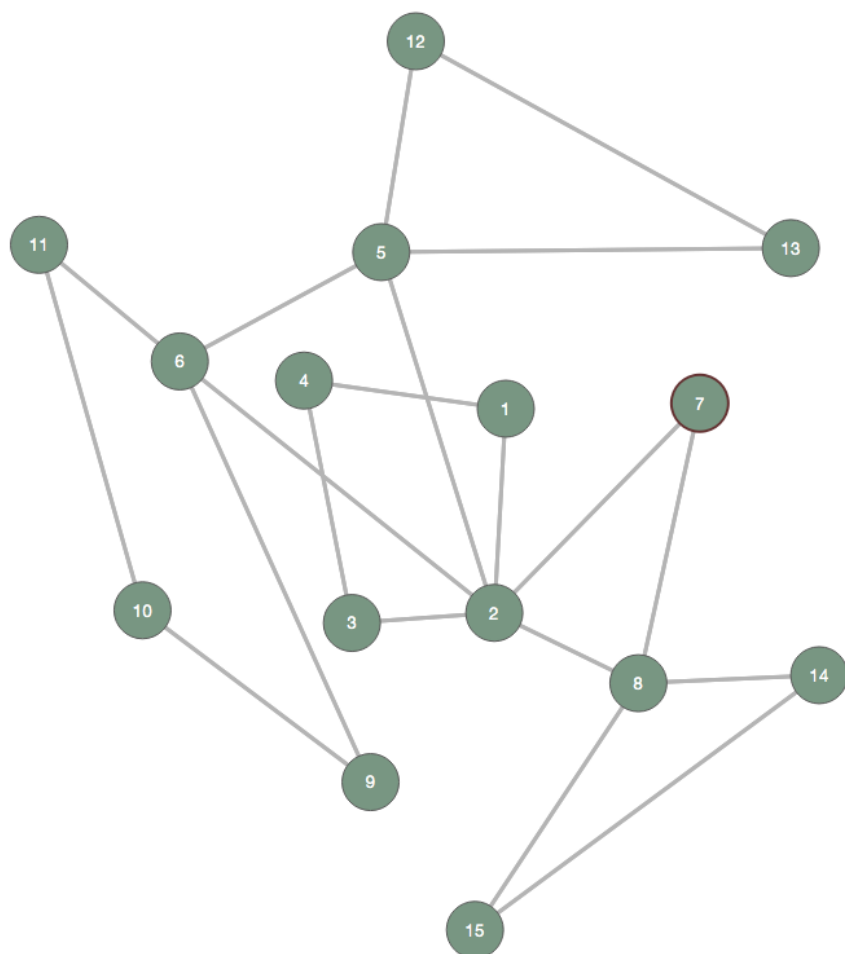
- vygeneruje se 10-15 vrcholů, které se postupně spojí hranami, dohromady tvořící jednu velkou cestu
- každému vrcholu se s pravděpodobností 2/5 přidělí jedna další náhodná hrana

Takto vygenerovaný graf bude vždy souvislý, a díky malému počtu hran i relativně přehledný. Graf je vždy generovaný jako neorientovaný. Pokud si uživatel přeje, může poté náhodně zorientovat hrany (Ctrl-R, D).

Generování Eulerovských grafů

Protože pro Eulerovské grafy musí platit, že každý vrchol má sudý stupeň, je také jednoduché nahlédnout, že musí ležet na nějaké kružnici. Generování grafu

Výsledný graf pak vypadá např. takto



Takto vygenerovaný graf má opět výhodu, že je díky menšímu počtu hran přehledný.

Rozmístění vrcholů

Při generování náhodného grafu jsou vrcholy vždy rozmístěny na spirálu, která se rozvíjí *zevnitř ven*. Pro grafy výše zmíněné náhodně generované grafy je toto rozložení relativně blízko tomu, co by si uživatel mohl představit, a stačí zpravidla pouze přemístit pár vrcholů uvnitř spirály, aby se příliš mnoho hran nekřížilo.

Algoritmy

Všechny algoritmy jsou implementované jako stavový automat, který se stiskem R přesune do počátečního stavu, a stiskem N postupně krokuje, až dojde do koncového stavu, kdy algoritmus doběhl.

Proto jsem zvolil zásobníkovou variantu DFS místo rekurzivní, aby šlo jednoduše ovládat průběh algoritmu.

DFS, BFS

Pro porovnání prohledávání do hloubky a do šířky je nejlepší zvolit stejný graf, a na něm pozorovat, jak se průběh jednotlivých algoritmů liší. Oba používají stejnou konvenci, a to že nenavštívený vrchol je tmavě zelený, otevřený je světle zelený a uzavřený je černý.

Jak DFS tak BFS umí pracovat s orientovanými grafy. Orientace hrany se změní označením hrany myší a stiskem O. Pro vrcholy A a B se postupně mění typ hrany na $A \rightarrow B$, $A \leftarrow B$, a $A \leftrightarrow B$.

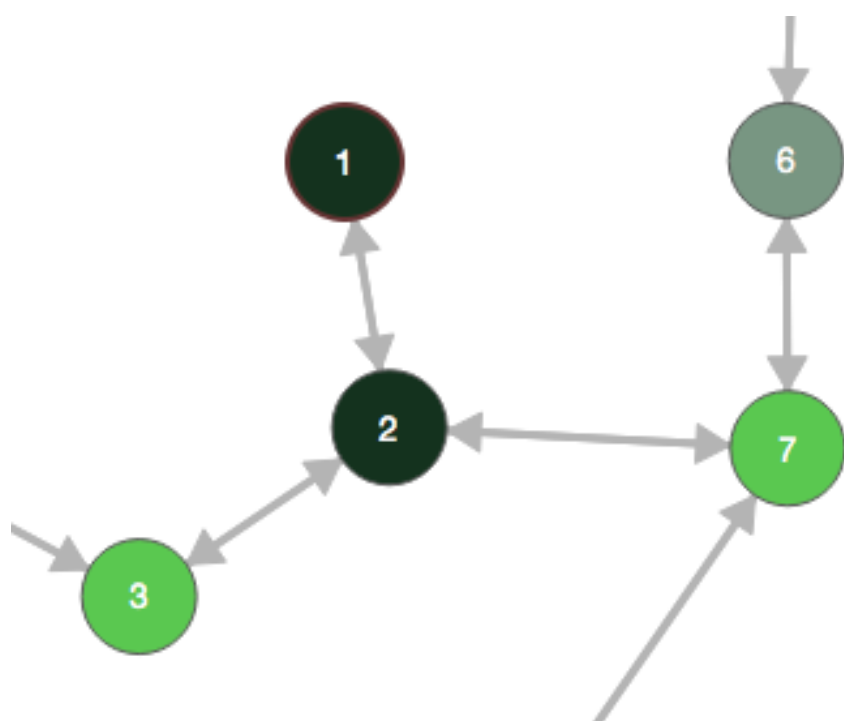


Figure 4: vertex-colors