

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Algoritmy

Dokumentace k implementaci tématu Obarvení grafu

10. prosince 2011

<u>Ševčík Radek</u>	[xsevc44]	40 %
Hykel Tomáš	[xhykel01]	40 %
Esmaildokht Mamaghani Amir Hossein	[xesmai00]	20 %

Obsah

Úvod.....	3
Tým a vývoj.....	3
Rozhraní.....	3
Algoritmus.....	4
Struktura kódu.....	5
Testování.....	5
Závěr.....	5
Použité zdroje.....	6

Úvod

Cílem tohoto dokumentu je přiblížit a popsat implementaci úlohy zaměřené na barvení neorientovaných grafů způsobem, při kterém žádné dva sousední uzly nesdílejí stejnou barvu. Program i s dokumentací vznikl jako týmový semestrální projekt do předmětu IAL. Použit byl jazyk C podle normy C99.

Tým a vývoj

Námi použitý vývojový cyklus:

1. Analýza požadavků a návrh
2. Definice rozhraní
3. Implementace
4. Testování
5. Dokumentace

Projekt se obešel i bez osobních schůzek, vystačili jsme si domluvou pomocí e-mailu a IM a použitím SVN repozitáře. Část týmu se na výsledném projektu nepodílela nijak, bylo to však podle jejich vlastního rozhodnutí.

Rozhraní

Program se po překladu příkazem *'make'* spouští souborem *ial11* a na vstupu přijímá jediný parametr, kterým je cesta k souboru s grafem; první řádek souboru obsahuje počet uzlů grafu a na dalších je definován samotný graf maticí sousednosti.

Výstupem programu je seznam uzlů seskupených podle toho, jaká jim byla přiřazena barva; v zájmu větší přehlednosti nejsou definovány konkrétní barvy (což je ve výsledku irelevantní), ale jen jejich pořadová čísla.

Vzorové užití:

```
$ ./ial11 tests/k3_3.txt
```

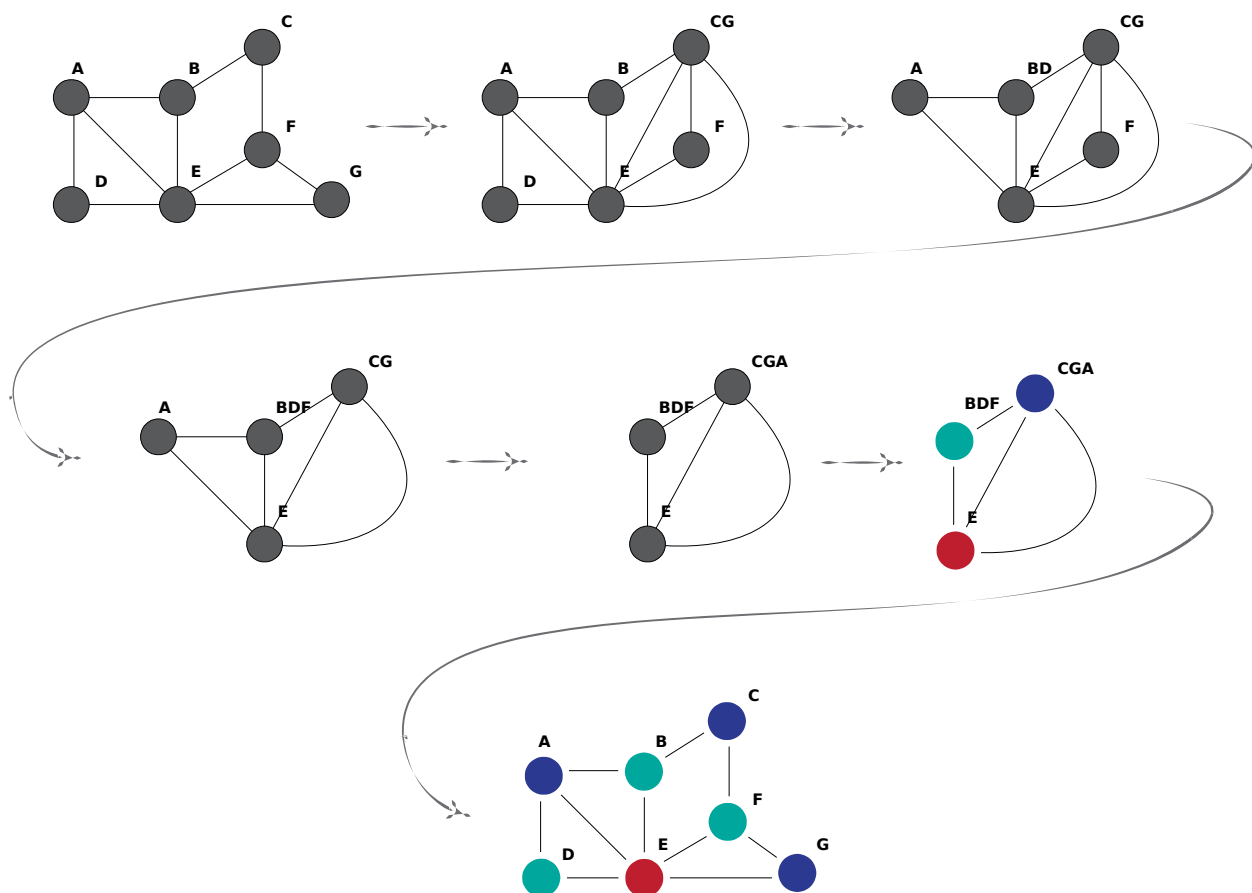
```
barva 1: 2 1 0
```

```
barva 2: 5 4 3
```

Algoritmus

Pro implementaci barvení matice jsme zvolili heuristický algoritmus, který využívá vlastností úplného grafu. Bývá nazýván “Barvení pomocí úplného grafu” nebo “Barvení slepováním vrcholů”.

Jeho princip spočívá v postupném sjednocování vrcholů obarvitelných stejnou barvou (tedy nesousedních) až do stavu, kdy se z původního grafu stane graf úplný. Vrcholům výsledného grafu se následně sekvenčně přiřadí samostatné barvy a provede se expanze do původního vztahu.



Teoretická složitost algoritmu je **exponenciální**; v nejhorším případě je při hledání dvou nespojených uzlů třeba projít celý graf.

Vybrali jsme si jej pro přijatelný kompromis efektivity a přímocarosti jeho implementace.

Struktura kódu

O zpracování vstupu a základní řízení programu se stará modul *ial11*.

Podstatnou funkcionalitu samotného barvení grafů zajišťují moduly *graph*, *matrix* a *vertex*. Každý z nich definuje vlastní struktury a operace nad nimi:

vertex pracuje se seznamem uzlů (vázaný seznam typu *vertex_t*),

matrix slouží pro operace s maticí sousednosti (struktura *matrix_t*) a

graph zastřešuje obě typy struktur (strukturou *graph_t*) a slouží k manipulaci s grafem jako takovým.

Podrobnější popis jednotlivých elementů je po spuštění '*make doc*' k dispozici v adresáři *doc/code*.

Testování

Při ověřování správné funkčnosti programu jsme použili řadu různých grafů a sledovali jsme jak správnost barvení, tak rychlost zpracování a využití systémových prostředků.

Experimentální ověřování asymptotické složitosti bylo prováděno na náhodně vygenerovaných grafech o počtu uzlů v řádů tisíců a víc. Tyto nám potvrdily, že průměrná složitost barvení s větším počtem uzlů narůstá exponenciálně.

V adresáři *test* jsou k dispozici vzorová testovací data + program *matgen* sloužící ke generování náhodných matic.

Vzorové výsledky testování náhodných grafů:

4000 uzlů:		10000 uzlů:		15000 uzlů:	
<i>real</i>	0m10.234s	<i>real</i>	2m21.587s	<i>real</i>	7m28.904s
<i>user</i>	0m10.189s	<i>user</i>	2m20.944s	<i>user</i>	7m27.561s

Závěr

Vývoj probíhal bez větších komplikací díky rozumnému návrhu datových struktur a volbě algoritmu vhodného pro transparentní implementaci. Díky tomu, že problematika barvení grafů hraje důležitou roli v mnoha výpočetních oblastech, si této zkušenosti rozhodně ceníme.

Použité zdroje

- GIBBONS, Alan M. *Algorithmic graph theory*. Cambridge : Cambridge University Press, 1985. 259 s.
- *Studijní opora předmětu FAST-HA53 (Teorie grafů)*