

Algorytmy Zaawansowane - kolorowanie $\Delta(G)$ dokumentacja końcowa

Albert Sadowski, Piotr Stanek
Wydział Matematyki i Nauk Informatycznych
Politechnika Warszawska

10 września 2013

1 Instrukcja użytkownika

Delta coloring jest aplikacją konsolową realizującą algorytm kolorowania grafu o czasie wykonania ograniczonym przez $\Delta(G)$.

Aplikacja przyjmuje jako argument graf ścieżkę do pliku zawierającego graf zdefiniowany w formacie DOT. Plik wykonywalny *deltaColoring* do standardowego wyjścia drukuje wynik realizacji algorytmu - graf w formacie DOT ze znacznikami odpowiadającymi za kolor poszczególnych wierzchołków grafu.

Przykładowe uruchomienie programu:

```
deltaColoring /path/to/input.dot
```

Skrypt *run.sh* uruchamia aplikację *deltaColoring* wraz z innymi programami odpowiadającymi za wygenerowanie pliku PDF z pokolorowanym grafem wyjściowym - rezultatem programu.

Przykładowe uruchomienie skryptu:

```
run.sh /path/to/input.dot
```

Skrypt korzysta z aplikacji *dot*, która rysuje graf podany w formacie DOT do formatu post script, oraz z aplikacji *ps2pdf*, która dla zadanego pliku w formacie post script generuje plik PDF. Skrypt automatycznie uruchamia program *evince* z wyjściowym grafem.

Wyjściowy plik PDF (*graph.pdf*) zapisany jest w folderze *output* wraz z plikiem *err.log* odpowiadającym za zapis strumienia *stderr*.

Przykładowe wejście - graf zdefiniowany w formacie DOT:

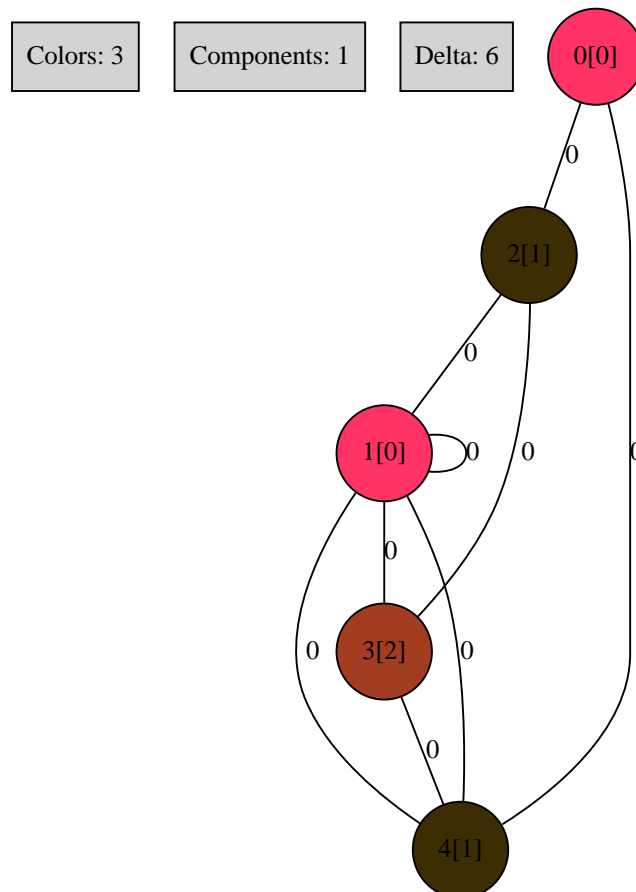
```
graph G {  
0;  
1;  
2;  
3;  
4;  
0--2 ;  
1--1 ;  
1--3 ;  
1--4 ;  
2--1 ;  
2--3 ;  
3--4 ;  
4--0 ;  
4--1 ;
```

```
}
```

Wywołanie programu *deltaColoring* z powyższym grafem zadany jako argument:

```
graph G {
node[shape="circle",style="filled"]
col[label="Colors: 3",shape="box"];
com[label="Components: 1",shape="box"];
del[label="Delta: 6", shape="box"];
0[fillcolor="#B88A00",label="0[0]"];
1[fillcolor="#B88A00",label="1[0]"];
2[fillcolor="#F5B800",label="2[1]"];
3[fillcolor="#FF6633",label="3[2]"];
4[fillcolor="#F5B800",label="4[1]"];
0--2 [label="0"];
1--1 [label="0"];
1--3 [label="0"];
1--4 [label="0"];
2--1 [label="0"];
2--3 [label="0"];
3--4 [label="0"];
4--0 [label="0"];
4--1 [label="0"];
}
```

Wywołanie skryptu *run.sh* z powyższym grafem zadany jako argument:



2 Zmiany względem dokumentacji wstępnej

Dokumentacja wstępna nie przewidywała formy/sposobu działania aplikacji. Ze strony algorytmicznej nie wprowadzono zmian.

3 Wnioski

Do implementacji zastosowano bibliotekę Boost Graph Library (BGL). Implementacja podstawowych algorytmów teorii grafów (tj. przeszukiwanie w głąb, algorytm Dijkstry) dostarczona wraz z BGL została wykorzystana.

4 Podział prac

- *Albert Sadowski* - kolorowanie grafu spójnego, spajanie rozwiązań cząstkowych (dokumentacja wstępna + implementacja), dokumentacja finalna.
- *Piotr Stanek* - znajdowanie wierzchołków rozspajających (dokumentacja wstępna + implementacja), testy.