Aleksandra Dolot
Marta Kuzak

PROJEKT: KOLOROWANIE KRAWĘDZI GRAFU

Sprawozdanie 3

I. Instrukcja obsługi aplikacji

Po uruchomieniu aplikacji, użytkownik z rozwijanego menu może wybrać trzy opcje: określić wejściowy plik tekstowy opisujący graf, wybrać algorytm, którym graf zostanie pokolorowany lub wyłączyć program. Do kolorowania grafu użytkownik wybrać może algorytm NC albo algorytm NTL. W wyniku działania algorytmu, krawędziom grafu przyporządkowane zostaną odpowiednie kolory oraz utworzony zostanie plik tekstowy w folderze *output*, z listą krawędzi i ich kolorów – format pliku będzie analogiczny do pliku wyjściowego, jedynie dodana zostanie zmienna określająca kolor.
Można również od razu wczytać cały katalog. Wtedy aplikacja przetestuje oba algorytmy (NC i NTL) wszystkimi znajdującymi się tam plikami.

Format plików wejściowych

Aplikacja tworzy grafy do kolorowania na podstawie plików tekstowych. Mają one następujący format:
1) W pierwszej linii pojawia się słowo kluczowe „VNUMBER”, a następnie liczba wierzchołków grafu. W ten sposób do grafu dołączane są wierzchołki o numerach wyrażonych kolejnymi liczbami naturalnymi od 0.
2) Kolejne linie definiują krawędzie grafu. Słowo kluczowe „EDGE” i potem kolejno: id krawędzi oraz pierwszy i drugi wierzchołek incydenty do niej.

Przykład pliku wejściowego przedstawiono poniżej:

```
VNUMBER 5
EDGE    0   0   1
EDGE    1   1   2
EDGE    2   2   3
EDGE    3   3   4
EDGE    4   1   3
```

Format plików wyjściowych

Po wykonaniu algorytmu aplikacja generuje pliki wyjściowe, które uzupełniają pliki wejściowe. Do każdej krawędzi dopisywany jest kolor krawędzi. Ponadto dodane są trzy parametry:

- COL_NUM : liczba wykorzystanych kolorów
- DEGREE : stopień grafu
- TIME : czas (mierzony w nanosekundach) wykonywania algorytmu

```
VNUMBER     5
EDGE  2     1     2     1
EDGE  1     0     1     2
EDGE  4     2     3     2
EDGE  3     1     3     3
EDGE  5     3     4     1
COL_NUM     4
DEGREE      3
TIME  94162
```

II. Opis kodu

Do sprawozdania dołączamy szczegółowo opisany w komentarzach kod programu. W celu realizacji algorytmów kolorowania krawędzi grafu utworzone zostały klasy opisujące krawędź (klasa *Edge*), Wierzchołek (klasa *Vertex*) i Graf (klasa *Graph*) oraz klasa MainFrame, w której stworzony został interfejs graficzny, funkcje wczytywania i zapisywania danych do pliku oraz z której wywoływane są testy.

Graf tworzony jest na podstawie pliku wejściowego. Do list wczytane są wierzchołki oraz krawędzie. Zaimplementowana została macierz sąsiedztwa opisująca połączenia w grafie – jej rzędy i kolumny odpowiadają kolejnym wierzchołkom, a wartości macierzy to kolory krawędzi, którymi wierzchołki są połączone (kolor 0 odpowiada brakowi krawędzi). Aby kontrolować wybrane kolory brakujące przypisane do poszczególnych wierzchołków, oraz zbiory kolorów brakujących poszczególnych wierzchołków utworzono szereg metod służących do ich aktualizacji. W szczególności każdy wierzchołek posiada HashMapę, przechowującą numery kolorów oraz zmienne logiczne, z której można odczytać czy dany kolor jest czy nie jest kolorem brakującym danego wierzchołka.

Algorytmy NC i NTL (oraz metoda Recolor – najważniejsza metoda algorytmu NTL) są metodami klasy *Graph*. Zaimplementowane zostały zgodnie z pseudokodami, które zostały zamieszczone w poprzednim sprawozdaniu. Algorytm NC, działający zachłannie, koloruje kolejne krawędzie najmniejszymi możliwymi kolorami. Algorytm NTL do pokolorowaniu krawędzi używa co najwyżej tyle kolorów, ile wynosi charakterystyka grafu plus 1, aby było to możliwe musi zmieniać kolory już pokolorowanych krawędzi, aby ostateczne pokolorowanie grafu było bliższe optymalnemu – cała logika algorytmu z tym związana znajduje się w metodzie Recolor. Metoda Recolor wywoływana jest, jeżeli jednej z krawędzi nie da się pokolorować, nie używając koloru większego niż ΔG+1, metoda przyjmuje za parametry wierzchołki, z którymi krawędź ta jest incydentna. Najważniejszymi, z implementacyjnego punktu widzenia, fragmentami metody Recolor jest tworzenie i rotowanie wachlarza od wierzchołka, z którego wychodzi krawędź, którą chcemy pokolorować, oraz konstrukcja i odwracanie ścieżek w grafie, dzięki, którym po przerotowaniu wachlarza, możemy pokolować wejściową krawędź. Zarówno ścieżka jak i wachlarz zaimplementowane zostały jako listy

zmiennych typu *Integer*, które określają numery wierzchołków, które znajdują się w ścieżce, bądź wachlarzu.

III. Testy

Program przetestowany został dla losowych grafów do 50 wierzchołków. Porównywane były czasy działania algorytmów NC i NTL oraz jakość ich działania, tzn. liczba kolorów, którymi krawędzie grafu zostały pokolorowane. Wnioski z testów dopiszemy później.

IV. Literatura użyta do rozwiązania problemu

1. M. Kubale, Optymalizacja dyskretna. Modele i metody kolorowania grafów, WNT, 2002
2. J. Wojciechowski, Grafy i sieci, Wydawnictwo Naukowe PWN, 2013
3. H. Gabow, T. Nishizeki, O. Kariv, D. Leven, O. Terada, Algorithms for Edge-Coloring Graphs, http://www.ecei.tohoku.ac.jp/alg/nishizeki/sub/e/Edge-Coloring.pdf
4. S. Nakano, X. Zhou, T. Nishizeki, Edge-Coloring Algorithms, http://www.ecei.tohoku.ac.jp/alg/nishizeki/sub/j/DVD/PDF_P/P032.pdf
5. X. Zhou, H. Suzuki, T. Nishizeki, An NC Parallel Algorithm for Edge-Coloring Series-Parallel Multigraphs, http://www.ecei.tohoku.ac.jp/alg/nishizeki/sub/j/DVD/PDF_J/J107.pdf