

Projekt z przedmiotu GIS - sprawozdanie I

Marek Jasiński, Przemysław Piotrowski

23 marca 2013

1 Treść zadania

Temat zadania Znajdowanie zbioru najkrótszych ścieżek pomiędzy parą wybranych lub wylosowanych wierzchołków grafu przy awaryjnych krawędziach.

Opis zadania Muszą być spełnione następujące warunki: a) pierwsza wygenerowana ścieżka: najkrótsza ścieżka pomiędzy wybraną parą wierzchołków b) każda następna wygenerowana ścieżka posiada krawędź różną od co najmniej jednej krawędzi ścieżki z punktu a) i minimalną długość. Krok b) jest powtarzany kolejno aż do wyczerpania wszystkich krawędzi ścieżki z punktu a) Dane do projektu: $G=(V,E)$ - graf o zadanej strukturze, wylosowana lub wybrana para wierzchołków. Zastosowanie: wyznaczanie ścieżek rezerwowych w sieci w sytuacji awarii pojedynczej krawędzi sieci. Literatura: N.Christofides, Graph Theory - an algorithmic approach, Academic Press 1975, str. 150- 157, 167-170. M.Sysło, N.Deo, J.Kowalik, Algorytmy optymalizacji dyskretnej, PWN 1995. N.Deo, Teoria grafów i jej zastosowania w technice i informatyce. PWN 1980.

2 Propozycja rozwiązania

Wejście programu Program będzie przyjmował na wejściu graf oraz dwa wierzchołki: źródłowy i docelowy. Graf będzie grafem nieskierowanym z ważonymi krawędziami (nieujemnymi). Zadany będzie w pliku tekstowym. Format tego pliku będzie zgodny z formatem DOT [1]. Przykładowe wywołanie programu może wyglądać w ten sposób:

```
$ ./ap graph.dot -s w1 -t w2
```

Przykładowy plik z zadaniem grafem wejściowym może wyglądać w ten sposób:

```
/* graph.dot */
graph G {
w1 -- w2 [weight=1];
w2 -- w3 [weight=1];
w1 -- w4 [weight=1];
w4 -- w2 [weight=1];
w2 -- w5 [weight=1];
w5 -- w3 [weight=1];
}
```

Przetwarzanie Graf będzie wewnętrznie reprezentowany w postaci listy sąsiedztwa, której implementację dostarcza biblioteka Boost Graph [2]. Pomimo, że biblioteka dostarcza implementację algorytmu znajdującego najkrótszą ścieżkę pomiędzy dwoma wierzchołkami, zaimplementujemy własny algorytm oparty na algorytmie Dijkstry. Ogólny algorytm będzie polegał na znalezieniu najkrótszej ścieżki pomiędzy wierzchołkami źródłowym i docelowym oraz wykorzystaniu jej do dalszego przetwarzania. Kolejne krawędzie z tej ścieżki będą tymczasowo usuwane, aby uruchamiać algorytm znajdowania najkrótszej ścieżki, który dostarczy alternatywną ścieżkę dla awaryjnej krawędzi. Warto zwrócić uwagę na to, że mogą istnieć krawędzie, których usunięcie spowoduje, że graf przestanie być spójny. Takie krawędzie zostaną oznaczone specjalnym kolorem, jako krytyczne. Nie ma alternatywnych ścieżek w grafie dla krawędzi krytycznych.

Wyjście programu Wizualizację rezultatu działania programu zapewni program NEATO z pakietu Graphviz [3]. Służy on do rysowania grafów nieskierowanych. Za pomocą programu NEATO zostaną wygenerowane pliki, na podstawie których zostaną wygenerowane strony HTML. Na stronach będzie można zobaczyć jak wyglądają najkrótsza ścieżka pomiędzy wierzchołkami źródłowym i docelowym oraz najkrótsze ścieżki alternatywne (po kliknięciu na krawędź, którą uznaje się za awaryjną).

3 Bibliografia

- [1] "The DOT Language"
<http://www.graphviz.org/doc/info/lang.html>

- [2] "The Boost Graph Library"
http://www.boost.org/doc/libs/1_53_0/libs/graph/doc/index.html
- [3] "Graphviz - Graph Visualization Software"
<http://www.graphviz.org/>