Ford-Fulkerson Przepływ maksymalny

Testowanie i weryfikacja oprogramowania

Czas	2012-12-12 1:00
Miejsce	Przyczółek
Obecni	MO

1 Wstęp

1.1 Opis lagorytmu

2 Instrukcja obsługi

2.1 Dane wejściowe

Algorytm Forda-Fulkersona opisany powyżej przyjmuje dwa argumenty:

- *FlowNetwork* - graf przepływu sieci
- *Search* - metoda wykorzystywana do wyszukiwania ścieżek powiększających.

2.1.1 Graf przepływu

Graf przepływu realizowany jest przy pomocy algorytmu z użyciem list powiązanych, *FlowNetworkAdjacencyArray*. Do każdego wierzchołka grafu przypisywane są dwie listy, krawędzi przednich oraz tylnich. Rozwiązanie to jest nieodpowiednie dla zastosowań z użyciem dużych grafów, ponieważ zajmuje spore ilości pamięci.
Argumenty wejściowe:

- *int* - ilość wszystkich węzłów w grafie
- *int* - indeks węzła początkowego
- *int* - indeks węzła docelowego
- *Iterator<EdgesInfo>* - lista krawedzi wraz z ich przepustowością

2.1.2 Krawędzie

Informacje o krawędziach grafu przechowywane są w kolekcji obiektów *EgdeInfo*, które zawierają informację o wierzchołkach, pomiędzy którymi dana krawędź się znajduje oraz o jakie posiada możliwości przepustowe.

- *int* - indeks węzła startowego
- *int* - indeks węzła końcowego
- *int* - wartość przepustowa krawędzi

2.2 Dane wyjściowe

3 Przykład użycia

W pierwszej kolejności ustalane są główne parametry grafu.

- Ilość wezłów grafu
- Indeks węzła źródłowego, *source*
- Indeks węzła docelowego, *sink*

```
int numVertices = 6;
int srcIndex = 0;
int sinkIndex = 5;
```

Następnie definiowane są poszczególne krawędzie grafu wraz z ich przepustowością.

- Indeks węzła początkowego krawedzi
- Indeks węzła końcowego krawędzi
- Maksymalna prapustowość krawędzi

```
ArrayList preIterator = new ArrayList();
EdgeInfo edge1 = new EdgeInfo(0, 1, 3);
EdgeInfo edge2 = new EdgeInfo(1, 3, 2);
EdgeInfo edge3 = new EdgeInfo(3, 5, 3);
EdgeInfo edge4 = new EdgeInfo(1, 4, 2);
EdgeInfo edge5 = new EdgeInfo(0, 2, 2);
EdgeInfo edge6 = new EdgeInfo(2, 4, 3);
EdgeInfo edge7 = new EdgeInfo(4, 5, 2);
EdgeInfo edge8 = new EdgeInfo(2, 3, 2);
preIterator.add(edge1);
preIterator.add(edge2);
preIterator.add(edge3);
preIterator.add(edge4);
preIterator.add(edge5);
preIterator.add(edge6);
preIterator.add(edge7);
preIterator.add(edge8);

Iterator<EdgeInfo> edges = preIterator.iterator();
```

Tworzony jest obiekt reprazentujący cały graf przepływu.

```
FlowNetworkAdjacencyList network = new FlowNetworkAdjacencyList(numVertices, srcIndex, sinkIndex, edges)↩
    ;

System.out.println(network.toString());
```

Wybierana jest funkcja wyszukująca i wykonywany jest algorytm Forda-Fulkersona.

```
//algorytm przeszukiwania wszerz
BFS_SearchList search = new BFS_SearchList(network);

//algorytm FORD–FULKERSON
FordFulkerson fordFulkerson = new FordFulkerson(network, search);

//wykonaj algorytm
fordFulkerson.compute();
```

Wyświetlenie wyznaczonego maksymlnego przepływu.

```
System.out.println("Wynik:");
System.out.println(network.toString());
```

4 Słownik pojęć i definicje