

Praktikum 3

Abgabe von :

- Moritz Spindelhirn
- Marjan Bachtari

Quellenangaben:

Es wurden keine Fremdquellen verwendet.

Der EdmondsKarp Algorithmus wurde anhand des englisch sprachigen Wikipedia Artikels umgesetzt.

http://en.wikipedia.org/wiki/Edmonds–Karp_algorithm

Bearbeitungszeitraum:

- 2 Stunden (Refactoring)
- 5 Stunden (FordFulkerson)
- 1 Stunden (EdmondsKarp Recherche)
- 3 Stunde (EdmondsKarp Implementation)
- 3 Stunden (Tests und Bugfixes)
- 1 Stunde (Grapherstellung - Netzwerk)
- => Zum Praktika 15 Stunden

Ford-Fulkerson:

Für die Implementierung des Ford-Fulkerson Algorithmus wurden die Klassen `FordFulkersonEdge` und `FordFulkersonVertex` erstellt.

Die Klasse `FordFulkersonEdge` enthält als zusätzliche Information den Flusswert.

Die Klasse `FordFulkersonVertex` leitet von der Klasse `Vertex` ab und enthält zusätzlich zu der bereits bestehenden Liste an Knoten zu denen eine ausgehende Verbindung besteht auch zusätzlich eine Liste mit

Knoten die eine eingehende Verbindung haben. Zudem werden Informationen für die Markierung in dieser Klasse gespeichert.

Ablauf

Nachdem der Startknoten markiert und in eine Liste der markierten Knoten eingefügt wurde wird so lange über jene Liste iteriert, bis keine weiteren Element vorhanden sind.

Bei jeder iteration werden die Nachbarn des ausgewählten markierten Knotens markiert sollte dies noch nicht der Fall sein und der Fluss noch nicht die Kapazitätsgrenze erreicht hat.

Wurde in dem Iterationsschritt der Zielknoten markiert wird die Flussoptimierung durchgeführt und die Markierungen, bis auf den Startknoten zurückgesetzt und die Iteration startet wieder.

Edmonds-Karp:

Die Klasse `EdmondsKarp` erbt von `FordFulkerson`, da es sich um eine erweiterte Implementierung handelt.

In dieser Implementierung wird in jeder Iteration ein vollständiger Weg vom Start zum Ziel gesucht und optimiert. Die Abbruchbedingung für die Schleife ist dadurch definiert, dass kein weiterer Weg zum optimieren gefunden wurde.

Der Weg wird in dieser Implementierung rekursiv über die Knoten gesucht. Hier wird entsprechend des Algorithmus immer der Weg mit der kleinsten zur Verfügung stehenden freien Kapazität genutzt.

Vergleich:

Für 10 Durchläufe mit einem Graphen mit 800 Knoten und 100.000 Kanten folgender Vergleich ergibt sich für die durchschnittliche Laufzeit.

```
Start:
.....
Done!
AVG time for FordFulkerson : 667ms
AVG time for EdmondsKarp : 204ms
```

Netzwerkerstellung:

Bei der Erstellung eines Netzwerks wird zunächst bei der Ertsellung ein Parameter mitgegeben, der dafür sorgt das der erstellte Graph zusammenhängend ist.

Danach kann man über einen weiteren Methodenaufruf an der Genratorenklasse die Quelle und Senke an den Graphen anhängen lassen. Die Anzahl der Kanten bezieht sich auf den Ursprünglichen Graphen. Beim anhängen der Quelle und Senke werden bis zu 10 neue Kanten eingefügt. Bei großen Graphen sollte diese Abweichung jedoch zu keinem ernsthaften Faktor werden.