

Team: 2_1: Sebastian Diedrich, Tim Hagemann

Aufgabenaufteilung:

1. Sebastian Diedrich : FordFulkerson Algorithmus & EdmondKarp Algorithmus
2. Tim Hagemann : Netzwerk-Generierung & Alternative Wege (BFS)

Quellenangaben:

[1] Graphentheorie und Operations Research, von C. Klauck und C. Maas

[2] <http://www.cse.unt.edu/~tarau/teaching/AnAlgo/Edmonds%E2%80%93Karp%20algorithm.pdf>

Bearbeitungszeitraum: 20.11.14 – 03.12.14

Aktueller Stand: Aufgabe vollständig implementiert.

(a) Welcher Algorithmus ist schneller?

Allgemein ist der Edmond-Karp Algorithmus schneller, es gibt aber auch Instanzen, wo Ford-Fulkerson ein schnelleres Ergebnis findet (wenn er „glücklich“ die Pfade wählt):

(b) Was haben Sie unternommen, um eine bessere Laufzeit zu erhalten?

1) Ford-Fulkerson:

Noch nichts

2) Edmond-Karp:

Abbruch eines Pfades, sollte Kante mit bereits gesättigtem Flow auftauchen.

(c) Lässt sich die Laufzeit ihrer Implementierung durch andere Datenstrukturen verbessern?

An manchen Stellen würde es sich anbieten eher eine Queue zu benutzen statt einer Liste, da gegebenenfalls nur der Zugriff auf den Anfang bzw. das Ende wichtig ist und hier die Queue um einiges schneller wäre.

(d) Was passiert, wenn Sie nicht-ganzzahlige Kantengewichte wählen?

Theoretisch macht es keinen Unterschied, da das Minimum (theoretisch) genauso funktioniert bei ganzzahligen wie bei nicht-ganzzahligen Zahlen. Jedoch ist bei der Implementation von Gleitkomma Zahlen das nicht so der Fall, da die meist dann nicht genau 1.5 sind, sondern 1.50000000004 und würden somit das Ergebnis verfälschen.

(e) Was passiert bei negativen Kantengewichten?

Beide Algorithmen werden ein Ergebnis liefern, jedoch wird es wohl auftreten, dass zum Beispiel mehr von der Quelle aus geht als bei der Senke ankommt, oder anders herum. Somit würde am Ende nicht das richtige Ergebnis entstehen.