

Team: 2_1: Sebastian Diedrich, Tim Hagemann

Aufgabenaufteilung:

1. Sebastian Diedrich : FW Algorithmus und GUI-Matrix für FW
2. Tim Hagemann : GUI und Graphen-Generierung

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

Bearbeitungszeitraum: 30.10.14 - 13.11.14

Aktueller Stand: Aufgabe vollständig implementiert.

(a) Bekommen Sie für den Graphen immer den gleichen kürzesten Weg?

1) Floyd-Warshall:

Wir erhalten auf jeden Fall immer den gleichen kürzesten Weg, da wir die Matrizen deterministisch erstellen und somit inhaltlich immer auf das gleiche Ergebnis kommen.

2) Dijkstra:

Nicht immer den gleichen Weg, aber immer den kürzesten Weg. Da wir beim Erstellen der Matrix uns die Reihenfolge aller Nachbarknoten "frei" auswählen können.

(b) Was passiert, wenn der Eingabegraph neg. Kantengewichte hat?

1) Floyd-Warshall:

So lange keine Kreise mit negativen Kantengewichten vorliegen, finden wir immer noch den kürzesten Weg.

2) Dijkstra:

Algorithmus findet einen Weg, der nicht mit Sicherheit der kürzeste ist.

(c) Wie allgemein ist Ihre Konstruktion von BIG?

Da der Algorithmus zur Erstellung eine Zufallszahl für Kanten, Knoten und Gewichtungen benutzt, ist es gegeben, dass nach einer bestimmten Anzahl von Durchgängen, alle möglichen Graphen generiert wurden. Ob der Graph gerichtet oder ungerichtete ist, wird vorher vom Benutzer entschieden.

(d) Wie testen Sie für BIG, ob ihr Algorithmus den kürzesten Weg gefunden hat?

Wir gehen davon aus, dass unsere Implementierungen der Algorithmen korrekt sind. Wir suchen uns zufällig ein Knotenpaar aus, beide Algorithmen müssen in dem Punkt „Kanten-Gesamt-Gewicht“ gleich sein. Der Weg kann sich allerdings unterscheiden.

(e) Wie müssen Sie ihre Lösung erweitern, um die Menge der kürzesten Wege zu bekommen?

Wir müssen für beide Algorithmen bei den Vorgängern in der Tabelle eine Menge eintragen, die alle gleichwertigen (gleiche Weggewichtung) Vorgänger beinhaltet. Bei der Erstellung des Pfades würden wir für jedes Element der Menge einen neuen Pfad erstellen.

(f) Wie müssten Sie ihre Lösungen erweitern, damit die Suche nicht-deterministisch ist?

1) Floyd-Warshall:

Im Algorithmus gibt es drei Möglichkeiten, die Reihenfolge der „Abarbeitung“ aller Knoten zu verändern. (z.B. $j = 1, 2, 3$ bei $|V| = 3$, als $j = 2, 1, 3$ abarbeiten). Da der Algorithmus nur die Vorgänger ändert, wenn sich ein kürzerer Weg zeigt, können so unterschiedliche Wege mit gleicher Weg-Gewichtung entstehen.

2) Dijkstra:

Algorithmus laut Definition nicht deterministisch. Siehe (a) 2.