Fortgeschrittene Algorithmen und Datenstrukturen

Seminar 3

1. Dijkstra's Algoritm

Der Algorithmus von Dijkstra berechnet den Spannbaum mit den kürzesten Pfaden, ausgehend von einem gegebenen Knoten v, zu allen anderen Knoten.

Implementieren Sie den Algorithmus in der Graph-Klasse aus Seminar 1.

Überlegen Sie dabei, wo und wie sie die berechneten Längen und Vorgängerknoten am besten speichern: In einer eigenen Liste oder direkt bei dem Knoten im Graphen. Begründen Sie die Antwort.

Welchen Aufwand hat der Dikstra-Algorithmus?

Kann der Dijkstra-Algorithmus früher abgebrochen werden, wenn nur der Pfad zu einem bestimmten Knoten gesucht wird? Wenn ja, wann?

2. Erweiterung um minimalen Spannbaum

Erweitern Sie die Implementierung aus 1, sodass der minimale Spannbaum mit dem Algorithmus von Prim berechnet wird. Es soll dabei möglichst wenig Codeverdopplung geben.

Berechnen Sie den minimalen Spannbaum für verschiedene Graphen.

3. Kruskal's Algorithm

Implementieren Sie den Algorithmus von Kruskal zur Bestimmung des minimalen Spannbaums. Achten Sie auf Effizienz bei der Entwicklung der *Disjoint Sets*. Zur Kantensortierung eignet sich der Einsatz einer `PriorityQueue` – oder zur Effizienzsteigerung ein *Fibonacci Heap* (der allerdings nicht in den Java Standardklassen zu finden ist).