

Writeup Exercise 10

Arianova Mendioroz (s0518061)
Bernhard Johannes Eiling (s0535003)

Finger exercises:

1.

Methoden und Signaturen des WeightedGraph sind:

void shortestPath()

um den kürzesten Weg durch den Graph zu finden

void cheapestPath()

um den günstigsten Weg durch den Graph zu finden

int getNodeDist(**int** node1, **int** node2)

um den Abstand zwischen zwei benachbarten Knoten zu finden

int getMinimumDist()

um die Vorgänger und die Distanz eines Knoten zum Start zu ermitteln und erneuern

private int[][] generateMatrix(**int** vert, **int** edge)

erzeugt eine Matrix mit zufällig gewählten Kantengewichten

int getValue(**int** row, **int** col)

liest die Werte der Adjazenzmatrix (für Testzwecke)

2.

Um den günstigsten Weg durch einen Graphen zu finden bietet sich der Dijkstra-Algorithmus an.

Um den kürzesten Weg durch einen Graphen zu finden kann man ebenfalls diesen Algorithmus verwenden. Nur muss man dann vorher alle Kantengewichte auf 1 setzen.

3.

Der Algorithmus wurde nicht in einer eigenen Datenstruktur angelegt sondern als Methode des Graphen. Die Adjazenzmatrix ist hier ein zweidimensionales Array das entsprechend der Anzahl der Knoten angelegt wird.

4.

Um zufällige Zahlen in Java zu erzeugen kann man entweder einen Random() Generator erzeugen und hier mittels nextInt() zufällig generierte Zahlen auslesen oder man verwendet die random() Methode von java.lang.Math. In dieser Übung wird zweites verwendet.

Um zufällige Zahlen von 0 bis i zu erzeugen benutzt man diesen Code:

int example = (**int**) (random() * i);