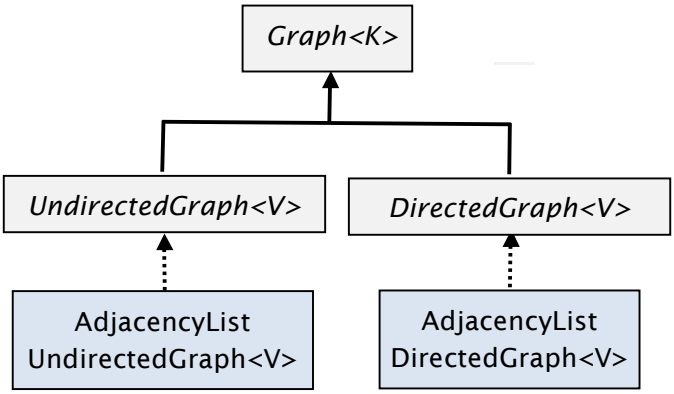
Algorithmen und Datenstrukturen
Angewandte Informatik
WS 2014/15
Prof. Dr. Oliver Bittel

HOCHSCHULE KONSTANZ
TECHNIK, WIRTSCHAFT UND GESTALTUNG

Aufgabenblatt 2



Auf der Web-Seite finden Sie Java-Interfaces für ungerichtete und gerichtete Graphen mit einer Javadoc-Dokumentation. Machen Sie sich damit vertraut.

Lösen Sie folgende Teilaufgaben:

1. Realisieren Sie eine Java-Klasse AdjacencyListUndirectedGraph für ungerichtete Graphen. Die Klasse benutzt eine doppelte HashMap, die jedem Paar von Knoten ein Gewicht als double-Wert zuordnet (siehe auch Skript Seite 3-27):

   ```
   HashMap<V,HashMap<V,Double>> adjacencyList;
   ```

2. Realisieren Sie eine Java-Klasse AdjacencyListDirectedGraph für gerichtete Graphen. Die Klasse verwendet sowohl für die Vorgänger- und als auch die Nachfolgerbeziehung eine doppelte HashMap.

3. Schreiben Sie eine Java-Klasse GraphTraversion mit folgenden generischen statischen Methoden:

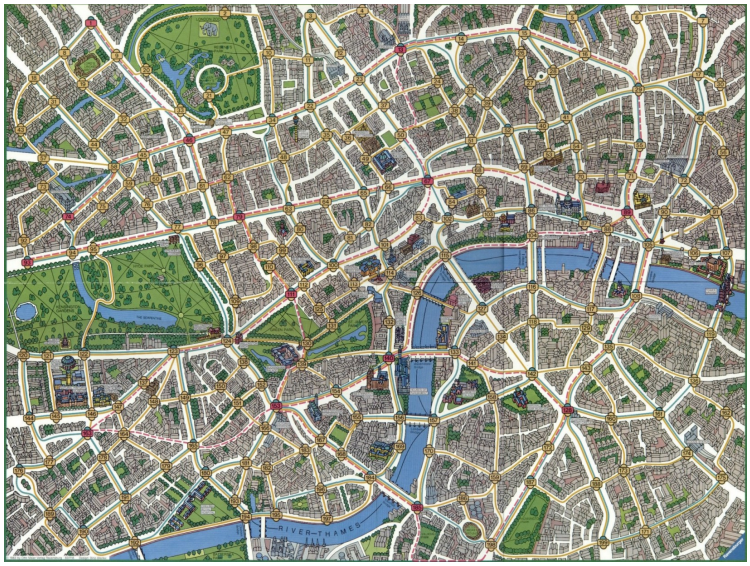
   - <V> List<V> depthFirstSearch(Graph<V> g, V s)
     liefert eine vom Knoten s gestartete Tiefensuche als Liste zurück.
   - <V> List<V> breadthFirstSearch(Graph<V> g, V s)
     liefert eine vom Knoten s gestartete Breitensuche als Liste zurück.
   - <V> List<V> topologicalSort(DirectedGraph<V> g)
     liefert eine topologische Sortierung des gerichteten Graphen g als Liste zurück.

4. Testen Sie die Tiefen- und Breitensuche an einem kleinen Graphen.

5. Definieren Sie einen Vorranggraphen (gerichteten Graphen), der das morgendliche Anziehen im Winter mit folgenden Tätigkeiten beschreibt: Strümpfe, Schuhe, Hose, Unterhose, Unterhemd, Hemd, Gürtel, Pullover, Mantel, Schal, Handschuhe und Mütze anziehen. Generieren Sie eine korrekte Anziehreihenfolge durch topologische Sortierung.

6. Der Spielplan des Spiels „Scottland Yard“ (Ravensburger; Spiel des Jahres 1983) besteht aus einer Menge von 199 Knoten (Punkten) in London, die durch Taxi, Bus oder U-Bahn verbunden sind. Auf der Web-Seite finden Sie einen Spielplan als jpg-Datei, eine Textdatei mit allen Kantenverbindungen und eine Klasse zur Animation des Spielplans mit farblichem Markieren der Knoten und der Verbindungen.

Lesen Sie aus der Textdatei nur die Taxi-Verbindungen ein und bauen Sie daraus einen ungerichteten Graphen. Animieren Sie die Tiefen- und die Breitensuche mit der vorhandenen Animationsklasse.



7. Schreiben Sie eine Klasse DijkstraShortestPath mit folgenden Methoden:

   - DijkstraShortestPath(Graph<V> g)
     Legt eine neue Instanz an zum Finden von kürzesten Wegen im Graphen g.
   - boolean searchShortestPath(V s, V g)
     Sucht einen kürzesten Weg von Startknoten s nach Zielknoten g. Der Dijkstra-Algorithmus soll abgebrochen werden, sobald der Zielknoten g erreicht wurde. Liefert true zurück, falls ein Weg gefunden wurde.
   - List<V> getShortestPath()
     Liefert einen kürzesten Weg zurück. Setzt eine zuvor erfolgreich durchgeführte Suche mit searchShortestPath(s,g) voraus.
   - double getDistance()
     Liefert die Länge eines kürzesten Weges zurück. Setzt eine zuvor erfolgreich durchgeführte Suche mit searchShortestPath(s,g) voraus.

8. Testen Sie die Klasse DijkstraShortestPath mit dem Scottland-Yard-Spielplan. Nehmen Sie für die Knotenverbindungen folgende Gewichte an: U-Bahn = 5, Taxi = 3 und Bus = 2. Für ein paar wenige Knotenverbindungen gibt es unterschiedliche Beförderungsmittel. Wählen Sie das billigste Beförderungsmittel.

9. Animieren Sie die Suche searchShortestPath(s,g) mit der gegebenen Animationsklasse.