

13. Übungsblatt

Ausgabe: 26. Januar 2011 **Abgabe:** 02. Februar 2011, 10 Uhr

Die Bearbeitung in Zweiergruppen ist ausdrücklich erwünscht.

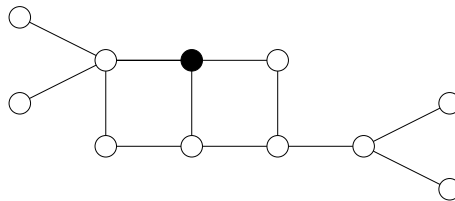
Aufgabe 1: DFS und BFS

10 Punkte

Führen Sie auf dem folgenden Graphen, beginnend beim ausgefüllten Knoten,

- (a) eine Tiefensuche,
- (b) eine Breitensuche

aus. Geben Sie jeweils die Knoten- und Kantennummern (analog zur Vorlesung) an.

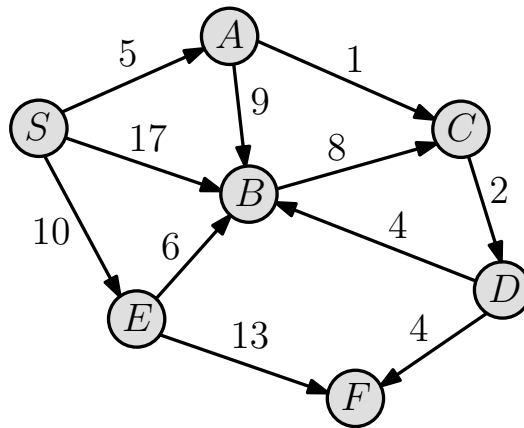


[Bitte wenden]

Aufgabe 2: Kürzeste Wege mit Dijkstra (optional)

5 Punkte

Bestimmen Sie für folgenden Graphen den Baum der kürzesten Wege ausgehend vom Startknoten S mit Hilfe des Algorithmus von Dijkstra, und bestimmen Sie die Distanzen aller Knoten von S . Dokumentieren Sie hierbei Ihre Schritte und geben Sie die Prioritätswarteschlange Q nach jedem Schleifendurchlauf der **while**-Schleife an.



Aufgabe 3: DFS und BFS

10 Punkte

Leiten Sie von der abstrakten Klasse `material.u13.AbstractGraph` eine Klasse `u13.name.Graph` ab, welche die abstrakten Methoden zur Tiefensuche und Breitensuche auf ungerichteten Graphen implementiert. Die zurückgegebenen Listen sollen die Knoten in der Reihenfolge enthalten, in denen sie durch die entsprechende Suche aufgefunden wurden, also in der Reihenfolge, in der die Knoten als *besucht* markiert werden. Die als Parameter übergebenen Wörterbücher sollen nach dem Durchlauf den Tiefensuch- bzw. Breitensuchbaum (repräsentiert durch Zeiger zum jeweiligen Elternknoten) und die Tiefensuch- bzw. Breitensuchnummern der Knoten enthalten.

Hinweise:

- Sie dürfen alle Klassen aus `java.util.*` verwenden. Insbesondere kann die Klasse `java.util.LinkedList` sowohl als Stack als auch als Warteschlange (*queue*) benutzt werden.
- Verwenden Sie für das Iterieren über Knoten und Kanten des Graphen, und über inzidente Kanten eines Knotens die Iteratoren aus `AbstractGraph`. Diese durchlaufen die jeweiligen Elemente in der Reihenfolge, in der sie im Graphen erzeugt wurden.