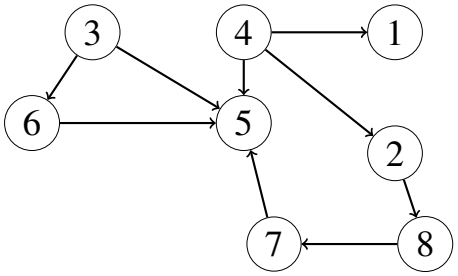
CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL
Institut für Informatik, Arbeitsgruppe Algorithmen und Komplexität
Prof. Dr. K. Jansen, K.-M. Klein, F. Land M. Rau

16. Juni 2016

Hausaufgaben zur Vorlesung »Algorithmen und Datenstrukturen«

Blatt 10

Hausaufgabe 10.1 (Topologisches Sortieren (3 Punkte))
Wenden Sie den Algorithmus aus der Vorlesung an um für folgenden gerichteten azyklischen Graphen eine topologische Sortierung anzugeben.



Hausaufgabe 10.2 (Bäume in Graphen (3 Punkte))
Sei $G = (V, E)$ ein ungerichteter Graph mit $|V| = n$ Knoten.
Beweisen Sie die Äquivalenz der folgenden Aussagen:

(a) G ist ein Baum

(b) G ist kreisfrei und $|E| = n - 1$

(c) G ist zusammenhängend und $|E| = n - 1$

Hausaufgabe 10.3 (Breitensuche (4 Punkte))
Es sei $G = (V, E)$ ein zusammenhängender ungerichteter Graph und es sei es $Vater(v)$ der Knoten $v' \in V$ durch den v in die Queue der Breitensuche hinzugefügt wurde. Zeigen Sie:

1. Der Graph $T = (V, E')$ mit $E' = \{(Vater(v), v) \mid v \in V, v \neq r\}$ ist ein gerichteter Baum mit Wurzel r, wobei r der Startknoten der Breitensuche ist.
2. Ist v ein Vorgänger von u in T, dann gilt $BFSNummer[v] < BFSNummer[u]$.
3. Jede Kante von e in $G = (V, E)$ verbindet Knoten deren Level sich um ≤ 1 unterscheidet.

Programmieraufgabe 10.4 (Priority Queue und Map)
Sie haben bereits Datenstrukturen kennen gelernt, die effizient Maps (Dictionarys) und Priority Queues (Prioritätswarteschlangen) umsetzen. Der abstrakte Datentyp *PQMap* vereint Aspekte von Map und Prioritätswarteschlange:

Wie eine Map enthält eine PQMap Schlüssel, und mit jedem Schlüssel ist genau ein Wert assoziiert. Ein Wert kann jedoch mit beliebig vielen Schlüsseln assoziiert sein (d.h. die Zuordnung von Schlüsseln zu Werten ist nicht notwendigerweise injektiv). Diese Assoziationen können abgefragt und verändert werden.

Zusätzlich fordern wir, dass die zulässigen Werte eine Quasiordnung bilden, d.h. für zwei Werte x und y gilt entweder $x \leq y$ oder $y \leq x$. (Aus $x \leq y$ und $y \leq x$ folgt aber nicht zwingend $x = y$. Ein Beispiel für eine solche Quasiordnung sind Strings mit der Festlegung $x \leq y \iff x.length \leq y.length$, für die $'aaa' \leq 'bbb'$ und $'bbb' \leq 'aaa'$ gilt.) Wie bei einer Prioritätswarteschlange erlaubt eine PQMap, einen bezüglich der Quasiordnung maximalen Wert abzufragen und zu entfernen.

Implementieren Sie den ADT PQMap in Java, der das gleichnamige Interface (siehe iLearn) umsetzt. Die Methoden `put`, `get`, `containsKey`, `size` und `remove` sollten sich wie die entsprechenden Methoden von `java.util.Map` verhalten, während `peek` und `poll` der Klasse `java.util.PriorityQueue` entnommen sind. Die Methoden `peekEntry` und `pollEntry` sollen sich wie `peek` und `poll` verhalten, jedoch zusätzlich den Schlüssel zurückgeben. Ihre Implementierung sollte für eine PQMap mit n Elementen die folgende durchschnittlichen Laufzeiten haben:

`get` $O(1)$
`containsKey` $O(1)$
`put` $O(\log n)$ (amortisiert)
`remove` $O(\log n)$ (amortisiert)
`size` $O(1)$
`peek` $O(1)$
`peekEntry` $O(1)$
`poll` $O(\log n)$ (amortisiert)
`pollEntry` $O(\log n)$ (amortisiert)

Hinweis Eine mögliche Implementierung wird im iLearn angedeutet und kombiniert Hashing mit einem Heap. Organisieren Sie Schlüssel/Wert-Paare in einem Max-Heap. Ordnen Sie zusätzlich mit einer HashMap jedem Schlüssel die Position des zugehörigen Eintrags im Heap zu.

Abgabe der theoretischen und praktischen Aufgaben Donnerstag, den 23. Juni, bis spätestens 14 Uhr. Die Abgabe der theoretischen Aufgaben erfolgt im Schrein. Die Abgabe der praktischen Aufgaben erfolgt im iLearn.