Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Matthias Baumgart

Wintersemester 2007/2008
Übungsblatt 11
11. Januar 2008

Effiziente Algorithmen und Datenstrukturen I

Abgabetermin: 18.01.2008 **vor** *der Vorlesung*

Aufgabe 1 (10 Punkte)

Geben Sie ein Verfahren an, das 5 Elemente mit 7 Vergleichen sortiert.

Aufgabe 2 (10 Punkte)

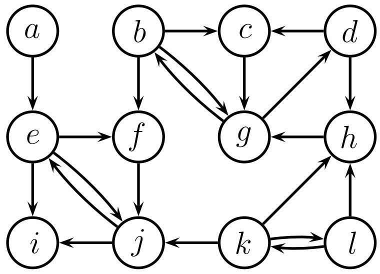
Sei ein Array $A[0..n-1]$ mit n paarweise verschiedenen natürlichen Zahlen gegeben. Ein Paar (i, j) heißt *Inversion* (bezüglich der aufsteigenden Sortierung), wenn $i < j$ und $A[i] > A[j]$ gilt.

Beschreiben und implementieren Sie einen Algorithmus, der die Anzahl der Inversionen in einem Array $A[0..n-1]$ in Laufzeit $O(n \log n)$ bestimmt. Die Eingabe A ist in einer Datei gegeben, deren Name beim Aufruf angeben wird. Die erste Zeile dieser Datei enthält die Anzahl n der Schlüssel. Jede weitere Zeile enthält genau einen Schlüssel.

Senden Sie Ihren Quelltext an `baumgart@in.tum.de`.

Aufgabe 3 (10 Punkte)

Führen Sie im folgenden gerichteten Graphen die Tiefensuche (DFS) aus.



Explorieren Sie dabei den Graphen entsprechend der *lexikographischen Ordnung* auf den Knotennamen, d.h. wenn Sie in einem Schritt unter mehreren Knoten wählen können, verwenden Sie den Knoten, der in der lexikographischen Ordnung vor allen anderen steht.

Beispiel: Angenommen man würde bei Knoten b starten, dann ist c der nächste betrachtete Knoten, da er unter allen Nachbarn von b (c, f und g) in der lexikographischen Ordnung der erste Knoten ist.

Aufgabe 4 (10 Punkte)

Ein ungerichteter Graph $G = (V, E)$ wird *bipartit* genannt, wenn es zwei nicht-leere Mengen V_1 und V_2 mit $V = V_1 \cup V_2$ und $V_1 \cap V_2 = \emptyset$ gibt, so dass für alle Kanten $\{u, v\} \in E$ entweder $u \in V_1$ und $v \in V_2$ oder $u \in V_2$ und $v \in V_1$ gilt.

Geben Sie einen möglichst effizienten Algorithmus an, der entscheidet, ob ein gegebener Graph bipartit ist, und analysieren Sie seine Laufzeit.