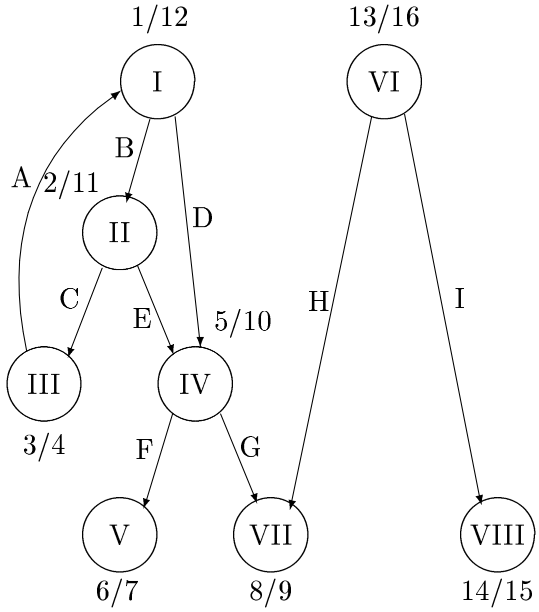
Technische Universität Chemnitz
Theoretische Informatik
Prof. Dr. Andreas Goerdt

Klausur Theoretische Informatik I – WS97/98
(keine Hilfsmittel zugelassen)

1. Aufgabe (2+4+3 Punkte)

a) Was verstehen Sie unter einer Kreuzkante bei der Tiefensuche?

b) Die nachfolgende Abbildung zeigt den Ablauf einer Tiefensuche über einem gerichteten Graphen mit den Kanten $A ... I$. Klassifizieren sie die Kanten nach dem Kantentyp.



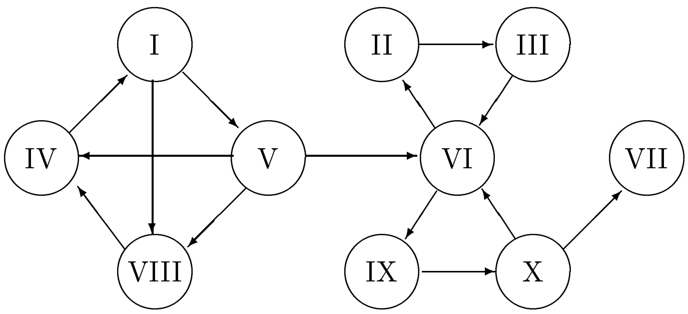
c) Wenn im vorherigen Graph die Kante A entfernt wird, dann ist der entstehende Graph topologisch sortierbar. Geben Sie eine topologische Sortierung dieses Graphen an.

2. Aufgabe (2+3+2+2 Punkte)

Zur Bestimmung von starken Zusammenhangskomponenten in gerichteten Graphen wird eine modifizierte Form der Tiefensuche verwendet.

a) Was sind starke Zusammenhangskomponenten in gerichteten Graphen?

b) Vollziehen Sie eine Tiefensuche über dem nachfolgenden gerichteten Graphen. Geben Sie dabei die Anfangs- und Beendezeiten an.



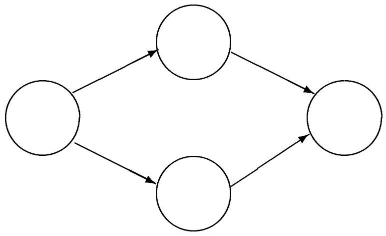
c) Beschreiben Sie kurz den Algorithmus zur Bestimmung von starken Zusammenhangskomponenten in Linearzeit.

d) Geben Sie die starken Zusammenhangskomponenten des Graphen an. In welcher Reihenfolge werden die starken Zusammenhangskomponenten bei Verwendung der Tiefensuche aus b) gefunden?

3. Aufgabe (3+2 Punkte)

a) Zwei Algorithmen zur Bestimmung von kürzesten Wegen in Graphen sind der Floyd-Warshall- und der Dijkstra-Algorithmus. Vergleichen Sie beide Algorithmen bezüglich Ein-, Ausgabeverhalten und Laufzeit. Gehen Sie auch auf den Einfluß der verwendeten Datenstrukturen ein.

b) Ergänzen Sie nachfolgenden Graphen so durch Kantengewichte, daß der Dijkstra-Algorithmus für mindestens einen Anfangsknoten falsche Ergebnisse liefert.



4. Aufgabe (2+3 Punkte)

a) Definieren Sie: Eine Funktion f ist von der Art $O(g)$, $(f, g : \mathbb{N} \rightarrow \mathbb{N})$.

b) Zeigen Sie, daß die Funktion $f(n) = an^3 + bn + c$, $(a, b, c > 0)$ von der Art $\Theta(n^3)$ ist.

5. Aufgabe (2+2+2+5 Punkte)

Wir betrachten den Sortieralgorithmus Bubblesort:

Eingabe: array A=$A[1...n]$ of integer
Ausgabe: Eine Sortierung der Elemente von A in A :
$A[1] \leq A[2] \leq ... \leq A[n]$

```
var sortiert: boolean;
sortiert:= false;
while sortiert=false do
      sortiert:= true;
      for i:=1 to n-1 do;
            if A[i] > A[i+1] then
                  vertausche A[i] und A[i+1]
                  sortiert:= false
```

a) Geben Sie alle Inhalte von A an, die der Algorithmus bei der Eingabe von $A = (4, 3, 1, 2)$ erzeugt.

b) Geben Sie für jedes n einen Inhalt von $A = (m_1, ..., m_n)$ an, auf dem der Algorithmus eine Zeit von $\Omega(n^2)$ benötigt.

c) Wie oft kann das größte Element während der Ausführung von Bubblesort höchstens bewegt werden? Begründen Sie Ihre Aussage.

d) Zeigen Sie die untere Schranke für das Sortieren von n Elementen: Jeder Sortieralgorithmus, der auf dem paarweisen Vergleich von Elementen basiert, hat im Worst Case eine Laufzeit von $\Omega(n \log(n))$.

6. Aufgabe (3+2 Punkte)

a) Betrachten Sie die *Union-Find*-Struktur ohne Heuristik. Geben Sie eine Folge von *Union-Find*-Operationen an, so daß die Folge beginnend mit der Partition $\{\{1\}, \{2\}, ..., \{n\}\}$ mindestens eine *Find*-Operation mit dem Laufzeitverhalten $\Omega(n)$ enthält.

b) Durch welche Heuristik ändert sich die Laufzeitkomplexität der *Find*-Operation auf $O(\log(n))$? Begründen Sie Ihre Antwort.