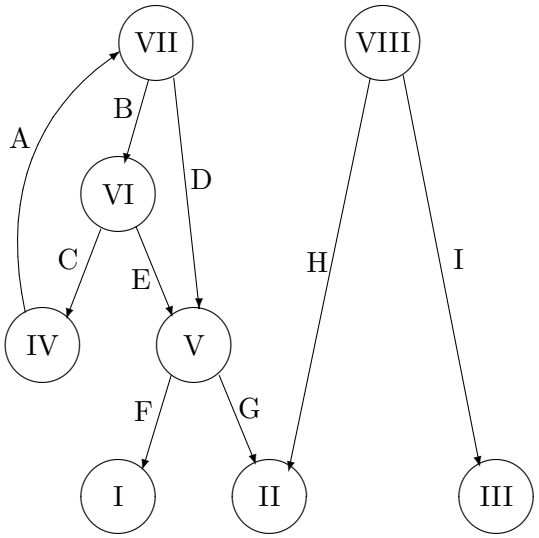
Technische Universität Chemnitz
Theoretische Informatik
Prof. Dr. Andreas Goerdt

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

1. Aufgabe (2+2+2+2+2 Punkte)

a) Vollziehen Sie ber dem nachfolgenden gerichteten Graphen eine Tiefensuche.
b) Was verstehen Sie unter einer Vorwrtskante bei der Tiefensuche?
c) Klassifizieren Sie entsprechend der Tiefensuche aus a) die Kanten $A...I$ nach dem Kantentyp.

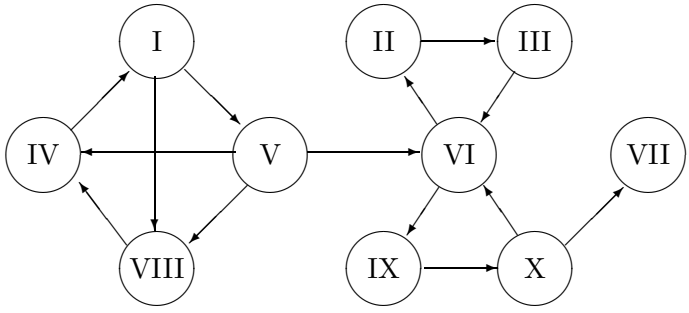


d) Wann ist ein Graph topologisch sortierbar?
e) Entfernen Sie aus dem vorigen Graphen eine Kante, so da dieser topologisch sortierbar wird. Geben Sie eine topologische Sortierung dieses Graphen an. Begrnden Sie Ihre Antwort.

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 ber dem nachfolgenden gerichteten Graphen. Geben Sie dabei die Anfangs- und Beendezeiten an.

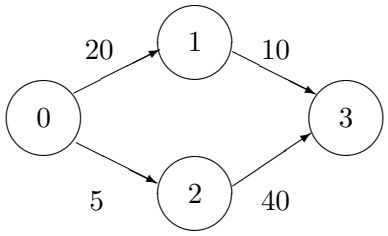


c) Beschreiben Sie kurz den Algorithmus zur Bestimmung von starken Zusammenhangskomponenten in Linearzeit.
d) Geben Sie die starken Zusammenhangskomponenten des Graphen an. In welcher Reihenfolge werden die starken Zusammenhangskomponenten bei Verwendung der Tiefensuche aus b) gefunden?

3. Aufgabe (3+2 Punkte)

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

b) Bestimmen Sie mittels Dijkstra-Algorithmus die krzesten Wege in folgendem Graphen. Startknoten sei dabei der Knoten 0.



4. Aufgabe (2+3 Punkte)

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

b) Gegeben sei eine Funktion $f(n) = an^3 + bn^2$, $(a, b, c > 0)$. Bestimmen Sie ein x, so da $f(n)$ von der Art $\Theta(n^x)$ ist. Beweisen Sie ihre Aussage.

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

Wir betrachten den Sortieralgorithmus Bubblesort:

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

```
var sortiert: boolean;
1 sortiert:= false;
2 while sortiert=false do
3       sortiert:= true;
4       for i:=n to 2 downto
5           if A[i] < A[i − 1] then
6               vertausche A[i] und A[i − 1]
7               sortiert:= false
```

a) Geben Sie alle Inhalte von A an, die der Algorithmus bei der Eingabe von $A = (T, H, E, O)$ erzeugt.

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

c) Wie oft kann das grte Element whrend der Ausfhrung von Bubblesort hchstens bewegt werden? Begrnden Sie Ihre Aussage.

d) Zeigen Sie die untere Schranke fr 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 (2+2 Punkte)

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

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