Technische Universität Chemnitz
Fakultät für Informatik
Prof. Dr. Andreas Goerdt

Chemnitz, den 03.06.2005

Klausur Theoretische Informatik I
WS 2004/2005 – Wiederholungsklausur

Aufgabe 1 (5+3 Punkte)

(a) Wir betrachten einen gerichteten Graphen G, der einen Weg von u nach v enthält. Eine Tiefensuche auf G ergibt $d[u] < d[v]$ für die Entdeckzeiten.

Geben Sie ein Beispiel dafür an, daß *trotz* der Voraussetzungen v nicht Nachfolger von u im Tiefensuchwald ist.

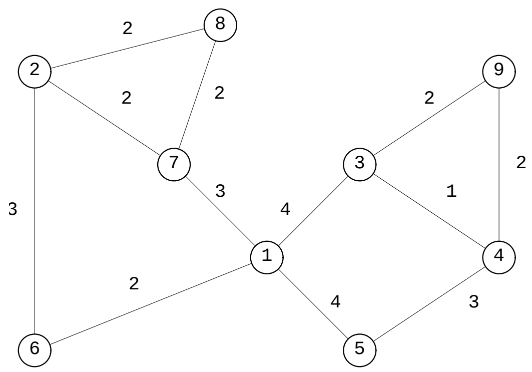
(b) Wir betrachten einen gerichteten, gewichteten Graphen mit Knotenmenge $V = \{1, \ldots, n\}$, wobei jede Kante die Kosten $= 1$ hat. Wir wollen das Distanzarray $D[1, \ldots, n]$ ermitteln, wobei

$$D[i] = \text{ die Distanz von Knoten } 1 \text{ nach } i.$$

sein soll. Geben Sie an, mit Hilfe welches Algorithmus der Vorlesung dieses Problem am effizientesten gelöst wird. Geben Sie auch seine Laufzeit an.

Aufgabe 2 (3+5 Punkte)

(a) Beweisen Sie, daß jeder minimale Spannbaum des folgenden Graphen die Kante $\{3, 4\}$ enthalten muß.



(b) Wir betrachten gerichtete, gewichtete Graphen mit Kosten ≥ 0 für jede Kante. Stellen Sie sich Dijkstras Algorithmus derart modifiziert vor, daß überall statt des Minimums das Maximum genommen wird. Ist die folgende Aussage wahr oder falsch:

Analog zum klassischen Dijkstra Algorithmus liefert der modifizierte Algorithmus die Längen der *längsten einfachen* Wege vom Startknoten ausgehend.

Beweisen Sie Ihre Antwort.

Aufgabe 3 (3+3+4 Punkte)

(a) Sei k eine beliebige Konstante (groß). Zeigen Sie, daß für alle hinreichend großen $x \ > \ x_0(k)$ gilt

$$x \ > (\log x)^k.$$

Verwenden Sie dabei die „Universalformel" der Vorlesung

$$2^{\varepsilon x} \ > \ x,$$

die bei konstantem $\varepsilon > 0$ für alle hinreichend großen x gilt.

(b) Zeigen Sie:

$$f(n) \ + \ g(n) \ \text{ ist } \ O(\max\{f(n)\,,\, g(n)\}).$$

Geben Sie die benötigte Konstante in der O-Notation *explizit* an.

(c) Zeigen Sie, daß gilt

$$\frac{2}{1 \, + \, (1/n)} \ = \ 2 \cdot \left(1 \ - \ \Omega\left(\frac{1}{n}\right)\right).$$

Geben Sie die benötigte Konstante in der Ω-Notation explizit an. Hinweis: Schreiben Sie den Bruch als „$2 \cdot \ \ldots$" und addieren Sie geschickt die 0 in geeigneter Darstellung. Alternativ und etwas komplizierter: Geometrische Reihe.

Aufgabe 4 (6+6 Punkte)

(a) Lösen Sie folgende Rekursionsgleichung genau:

$$T(0) = 2, \; T(1) = 10$$

$$T(n) = 10 \cdot T(n-1) \; - \; 16 \cdot T(n-2)$$

(b) Geben Sie die Rekursionsgleichung für die Laufzeit von Mergesort an. Gehen Sie dabei davon aus, daß die Feldlänge eine 2er-Potenz ist.

Lösen Sie die Gleichung und beweisen Sie die Korrektheit Ihrer Lösung durch Induktion.

Aufgabe 5 (5+5 Punkte)

(a) Betrachten Sie die beiden Worte

$$W = a\,b\,s\,a\,t\,u\,b\,v \text{ und } V = s\,s\,a\,u\,w\,v.$$

Geben Sie die längste gemeinsame Teilfolge (im Sinne der Definition der Vorlesung) von V und W an. Füllen Sie dazu die Tabelle auf dem Lösungsblatt so aus, wie es der (dynamische Programmier-)Algorithmus der Vorlesung zum Finden einer längsten gemeinsamen Teilfolge macht.

(b) Gegeben seien fünf Wörter $W_1, \ldots, W_5$ in alphabetischer Reihenfolge, sowie deren Wahrscheinlichkeiten (relative Häufigkeiten)

$$p_1 = \frac{3}{10}, \quad p_2 = \frac{1}{10}, \quad p_3 = \frac{2}{10}, \quad p_4 = \frac{3}{10} \text{ und } p_5 = \frac{1}{10}.$$

Welche Kosten (im Sinne der Vorlesung) hat der optimale statische binäre Suchbaum in diesem Fall?

Füllen Sie die Tabelle auf dem Lösungsblatt so aus, wie es der entsprechende (dynamische Programmier-)Algorithmus der Vorlesung machen würde.

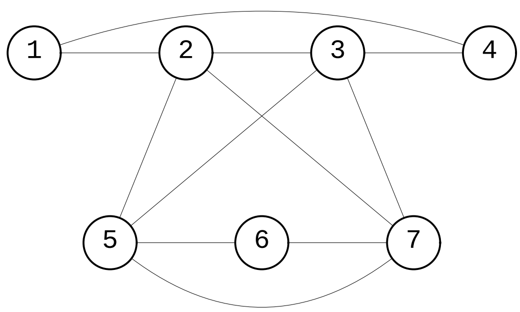
Aufgabe 6 (3+6 Punkte)

Wir betrachten ungerichtete, zusammenhängende Graphen.

(a) Geben Sie ein Kriterium an, das es erlaubt, effizient (d.h. in Polynomialzeit) zu erkennen, ob ein Graph einen Eulerschen Kreis hat oder nicht.

(b) In Vorlesung und Übung wurde ein Polynomialzeitalgorithmus zum Finden eines Eulerschen Kreises angegeben.

Demonstrieren Sie den Ablauf dieses Algorithmus an nachfolgendem Graphen. Geben Sie die wichtigsten Zwischenschritte dem Korrektor verständlich an!



Der Graph liege in Adjazenzlistendarstellung vor. Auf den einzelnen Adjazenzlisten treten die Knoten sortiert nach ihrem Namen (= Nummer) auf. Gehen Sie davon aus, daß der Algorithmus die Adjazenzlisten von links nach rechts bearbeitet. Der Algorithmus beginne auf Knoten 1. Betrachten Sie das Löschen einer Kante als gegebenes Primitiv.

Aufgabe 7 (6 Punkte)

Bestimmen Sie in dem folgenden Netzwerk den maximalen Fluß von Q nach S. Benutzen Sie dafür den Algorithmus von Edmonds und Karp.