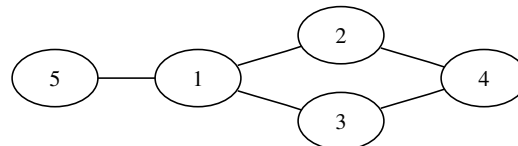


Klausur Theoretische Informatik I
Wintersemester 2005/2006
7 Aufgaben, Zeit 2 Stunden

1. Aufgabe

(2 + 3 + 3 = 8 Punkte)

- (a) Geben Sie die „Darstellung mittels zweier arrays“ des folgenden ungerichteten Graphen an.



- (b) Geben Sie an, wieviele Dreiecke es in einem *ungerichteten* Graphen mit n Knoten maximal gibt.
- (c) Wir betrachten *gerichtete* Graphen.

(c1) Geben Sie die Anzahl gerichteter Graphen mit Knoten 1, 2, 3 an, die die Anordnung

$$(3, 2, 1)$$

der Knoten als topologische Sortierung haben!

(c2) Geben Sie die Anzahl gerichteter Graphen mit Knoten 1, 2, 3, \dots , n an, die die Anordnung

$$(v_1, v_2, v_3, \dots, v_n)$$

als topologische Sortierung haben!

2. Aufgabe

(3 + 3 = 6 Punkte)

Beweisen Sie die folgenden Aussagen.

(a) $\frac{5}{3 - 7/n^2}$ ist $(5/3) + O(1/n)$.

(b) $2^{1+\Omega(\log n)}$ ist $n^{\Omega(1)}$.

3. Aufgabe

(3 + 3 = 6 Punkte)

- (a) Demonstrieren Sie das Linearzeitverfahren der Vorlesung bzw. Übung zur Erstellung eines heaps an folgendem array A von Schlüsselwerten. Der heap soll so angeordnet sein, daß der maximale Schlüsselwert an der Spitze steht. Stellen Sie die verschiedenen Zustände von A im Verlaufe des Algorithmus dar.

$$A = (1, 3, 5, 2, 7, 2, 10, 4, 11).$$

- (b) Beweisen Sie, daß das Verfahren gemäß (a) wirklich in Linearzeit läuft. Hinweis: Sie sollten auf die Funktion „Sicker“ Bezug nehmen.

4. Aufgabe

(2 + 2 = 4 Punkte)

- (a) Konstruieren Sie ein Flussnetzwerk, so daß gilt: Es gibt zwei Knoten a und b , so daß der Algorithmus von Edmonds Karp den Fluss zwischen a und b mehrmals erhöht, und zwar so:
- Zunächst wird $f(a, b)$ erhöht
 - dann $f(b, a)$
 - dann wieder $f(a, b)$.
- (b) Geben Sie eine (möglichst gute) obere Schranke dafür an, wie häufig Edmonds Karp $f(a, b)$ oder $f(b, a)$ erhöht. Mit kurzer Begründung (!)

5. Aufgabe

(1 + 5 = 6 Punkte)

- (a) Geben Sie die Anzahl der skalaren Multiplikationen an, die man braucht, um eine $m \times k$ Matrix mit einer $k \times l$ Matrix mit dem direkten Algorithmus zu multiplizieren! Mit (sehr) kurzer Begründung.
- (b) Seien Matrizen folgender Größe gegeben:

$$M_1 : 10 \times 20, M_2 : 20 \times 30$$

$$M_3 : 30 \times 5, M_4 : 5 \times 100$$

- (b1) Geben Sie die Größe des Matrixprodukts $M_1 \times M_2 \times M_3 \times M_4$ an!

- (b2) Demonstrieren Sie den auf dem dynamische Programmieren basierenden Algorithmus zur Ermittlung der minimalen Anzahl von Multiplikationen bei der Berechnung des Produkts $M_1 \times \dots \times M_4$!

Füllen Sie dazu die Tabelle auf dem Lösungsblatt aus. Beachten Sie: Die Multiplikation von zwei Matrizen wird hier mit dem „Trivialalgorithmus“ gemäß (a) gemacht.

6. Aufgabe

(2 + 3 = 5 Punkte)

- (a) Wir betrachten die folgende Belegung der aussagenlogischen Variablen x_1, x_2 (1 = wahr, 0 = falsch)

$$x_1 = 1, x_2 = 0.$$

Geben Sie eine Formel F in konjunktiver Normalform an, so daß die angegebene Belegung autark ist. Die Belegungen $x_1 = 1$ und $x_2 = 0$ *alleine* sollen allerdings beide *nicht* autark sein.

- (b) Demonstrieren Sie den Polynomialzeitalgorithmus für das 2-SAT-Problem an folgender Formel.

$$(a \vee b) \wedge (\neg a \vee b) \wedge (\neg a \vee \neg b) \wedge (x \vee y) \wedge (\neg x \vee \neg y) \wedge (z \vee y)$$

7. Aufgabe

(3 + 3 + 2 = 8 Punkte)

Wir betrachten den effizientesten (logarithmisches Kostenmaß) rekursiven Algorithmus der Vorlesung zur Multiplikation von großen Zahlen der Vorlesung für das 10'er System.

- (a) Demonstrieren Sie den Ablauf des Algorithmus für folgende Eingaben

1234 und 5678,

indem Sie den Prozeduraufrufbaum hinschreiben. Es reicht, an jedem Knoten des Baums die aktuellen Parameter anzugeben.

- (b) Schreiben Sie eine Summenformel für die Laufzeit des Algorithmus bei Eingabe von Zahlen mit n Ziffern, n Zweierpotenz, hin. Vereinfachen Sie die Summenformel soweit es geht, um die Laufzeit auszurechnen.
- (c) Geben Sie die # benötigter Speicherplätze des Algorithmus in etwa an, das heißt in der O -Notation.