14. Übung (Wiederholung)

Aufgabe 1: Divide & Conquer-Anwendung (2 + 1 = 3 Punkte)

Gegeben ist eine Permutation $\pi : \{1, \dots, n\} \to \{1, \dots, n\}$ (also eine bijektive Abbildung). Gesucht ist die Anzahl i_π von Inversionen der Permutation π, definiert durch:

$$i_\pi := |\{(i,j) \;\mid i < j, \pi(i) > \pi(j)\}|$$

(a) Geben Sie einen subquadratischen Algorithmus an (also Laufzeit $o(n^2)$, asymptotisch schneller als n^2), der zu Eingabe π die Anzahl i_π von Inversionen von π bestimmt und begründen Sie kurz die Korrektheit.

(b) Analysieren Sie die Laufzeit Ihres Algorithmus, wenn π als `integer`-Array `pi` mit `pi[i]`$= \pi(i)$ gegeben ist.

Aufgabe 2: Amortisierte Analyse (1 + 2 + 2 = 5 Punkte)

Eine Queue mit den Operationen **add** und **del** lässt sich auch mit zwei Stacks S und T implementieren. Die Operationen der Queue werden dann mit den Operationen der beiden Stacks wie folgt implementiert:

```
add(k):
  S.push(k)
```

```
del:
  if ( T.is_empty )
    while ( S.not_empty )
      T.push(S.pop)
  return(T.pop)
```

(a) Verdeutlichen Sie die Funktionsweise dieser Implementierung, indem Sie angeben, wie sich die beiden Stacks bei der Ausführung dieser Operationenfolge verändern:

add(1), **add**(2), **del**, **add**(3), **add**(4), **del**, **del**, **add**(5), **del**, **del**

(b) Wir nehmen an, dass **push** und **pop** jeweils den Aufwand 1 haben. Führen Sie mit der Bankkonto-Methode eine amortisierte Aufwandsanalyse der Operationen **add** und **del** durch, um zu zeigen, dass sie konstanten amortisierten Aufwand haben.

(c) Führen Sie nun eine amortisierte Aufwandsanalyse der Operationen **add** und **del** mit der Potential-methode durch.

Aufgabe 3: Datenstruktur mit Intervallabfragen (2 + 1 = 3 Punkte)

(a) Entwerfen Sie eine Datenstruktur, die eine Menge von Fließkommazahlen speichern kann und die folgende Operationen bietet:

add(x): Fügt die Zahl x zur Menge hinzu.

del(x): Löscht die Zahl x aus der Menge, falls vorhanden.

int(x,y): Gibt die Anzahl der gespeicherten Elemente z mit $x \leq z \leq y$ aus.

Jede dieser Operationen soll amortisierten Aufwand $O(\lg n)$ haben.

(b) Weisen Sie nach, dass Ihre Datenstruktur die amortisierte Aufwandsschranke einhält.

Bitte wenden!

Abgabe der Übung: Montag 12.02.2007, 16h00;
Abgaben in Papierform in Raum 05-230.

Johannes Gutenberg-Universität Mainz · Staudingerweg 9 · D-55099 Mainz
Internet: http://www.informatik.uni-mainz.de/lehre/dsea
E-Mail: schoemer@informatik.uni-mainz.de

Aufgabe 4: Topologische Sortierung (1 + 2 + 1 = 4 Punkte)

Wir betrachten einen Graphenalgorithmus, der für einen gerichteten, zyklenfreien Graphen (DAG) $G = (V, E)$ wie folgt vorgeht:
Bestimme einen Knoten v mit Eingrad 0, gebe v aus und entferne v aus G. Wiederhole das solange, bis alle Knoten ausgegeben wurden.

(a) Zeigen Sie, dass für die Ausgabe $v_1, v_2, \ldots, v_n$ des Algorithmus folgende Eigenschaft gilt:

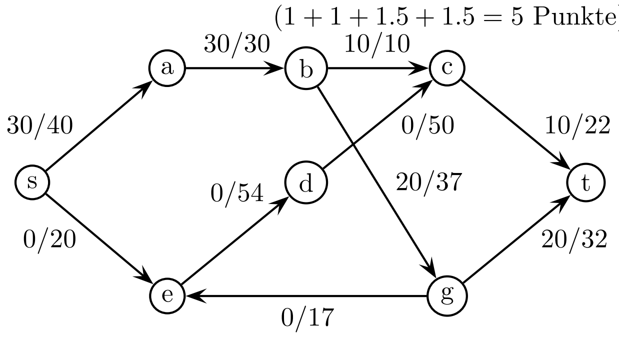
$$i < j \quad \Rightarrow \quad (v_j, v_i) \notin E$$

(b) Geben Sie eine genaue Implementierung dieses Algorithmus an, die als Gesamtlaufzeit $O(|V| + |E|)$ hat.

(c) Analysieren Sie die Laufzeit Ihres Algorithmus und weisen Sie die Einhaltung der vorgegebenen Laufzeitschranke nach.

Aufgabe 5: Maximaler Fluss (1 + 1 + 1.5 + 1.5 = 5 Punkte)

Wir betrachten das abgebildete Netzwerk G, das bereits einen Fluss f enthält. Zu jeder Kante ist ein Tupel angegeben, das im ersten Eintrag den Fluss durch diese Kante und im zweiten Eintrag die Kapazität der Kante angibt.



(a) Bestimmen Sie den Wert des Flusses.

(b) Finden Sie einen flussverbessernden Pfad für f und bestimmen Sie den erhöhten Fluss.

(c) Bestimmen Sie einen Schnitt mit minimaler Kapazität.

(d) Bestimmen Sie einen maximalen Fluss.

Aufgabe 6: Änderung maximaler Fluss (2 Punkte)

Sei f ein maximaler Fluss auf einem Netzwerk $G = (V, E)$ mit Kapazitätsfunktion $c : E \to \mathbb{N}$ und sei $k \in \mathbb{N}$, $k > 0$. Wir ersetzen die Kapazitätsfunktion c durch c', die für alle $e \in E$ definiert ist als

$$c'(e) := c(e) + k$$

Was können Sie über den Wert $|f'|$ des maximalen Flusses bezüglich Kapazitäten c' im Vergleich zum Wert $|f|$ aussagen?

Aufgabe 7: LP-Transformation (1, 5 + 1, 5 = 3 Punkte)

Wir betrachten folgendes lineares Programm:

$$\begin{array}{ll} \max x_1 + 2x_2 + 3x_3 & \\ \text{s.t.} & x_1 - x_2 \geq 10 \\ & 2x_1 + x_2 - x_3 \leq 20 \\ & x_1 + 2x_2 \geq x_3 \\ & x_1, x_2 \geq 0 \end{array}$$

(a) Transformieren Sie das LP in Ungleichungsstandardform!

(b) Transformieren Sie das LP in Gleichungsstandardform!