4. Übungsblatt für Track AI zur Vorlesung Entwurf und Analyse von Algorithmen, WS 13/14

Abgabe: *Bis* ***Freitag****, 15.11.2013, 12:00 Uhr, Kasten im Treppenhaus 48-6.*

Der Fehler der Woche

Aufgabe: Gib einen Algorithmus an, der in Zeit $\Theta(\log n)$ läuft.

Antwort:

```
a = x
while ( a <= n ) {
  a = x div a
}
```

Was ist falsch?

Basisaufgaben

B1: Asymptotiken

3 Punkte

Für Funktionen $f, g : \mathbb{N} \to \mathbb{R}_0^+$ sei die Relation $\prec$ definiert durch

$$f(n) \prec g(n) :\Longleftrightarrow f(n) \in o(g(n)) .$$

Seien nun $\varepsilon, c \in \mathbb{R}$ Konstanten mit $0 < \varepsilon < 1 < c$. Ordnen Sie die Funktionen

$$n^{\log(n)}, \quad \log(n), \quad 1, \quad c^{c^n}, \quad (c^c)^n, \quad n^c, \quad \log(\log(n)), \quad n^\varepsilon, \quad c^n \quad \text{und} \quad n^n$$

bzgl. $\prec$ an. Beweisen Sie die Korrektheit Ihrer Anordnung.

Hinweis: Können Sie bestimmte Eigenschaften von $\prec$ – die Sie natürlich erst nachweisen müssten – ausnutzen, um sich Arbeit zu sparen?

B2: Rekursionsgleichungen

3 Punkte

Ermitteln Sie Θ-Asymptotiken für die durch die folgenden Rekursionsgleichungen beschriebenen Zahlenfolgen! Treffen Sie dafür geeignete Annahmen an n und beweisen Sie Ihre Behauptungen.

a) $A(1) = 1,$

$$A(n) = \frac{5}{2} \cdot A\left(\frac{n}{2}\right) + \frac{n}{2}, \quad n \geq 2.$$

b) $B(1) = 6,$

$$B(n) = B\left(\frac{n}{3}\right) + n - 1, \quad n \geq 3\,.$$

c) $C(1) = 1,$

$$C(n) = 4 \cdot C\left(\frac{n}{2}\right) + 7 \cdot \left(\frac{n}{2}\right)^2, \quad n \geq 2\,.$$

B3: Speicherabbildungsfunktionen

3 Punkte

Finden Sie eine sequentielle Speicherabbildungsfunktion, welche zu einer beliebigen $n \times n$ Matrix eine Erweiterung zu einer $(n+k) \times (n+k)$-Matrix, $k \geq 1$, ohne Umstellung der vorhandenen Elemente zulässt, und in Zeit $\mathcal{O}(1)$ berechnet werden kann. Begründen Sie die Korrektheit Ihrer Lösung!

B4: Algorithmen auf Arrays

3 Punkte

Es sei eine $n \times n$-Matrix $A : [1:n] \times [1:n]$ gegeben, in der jedes Element $A[i,j]$ die Werte `Wasser` oder `Ente` annehmen kann. Zwei Enten gehören zu derselben Entenfamilie, wenn sie entweder in horizontal, vertikal oder diagonal benachbarten Feldern "schwimmen".

Entwerfen Sie einen Algorithmus, der Folgendes leistet: Bei Eingabe (i,j), $1 \leq i,j \leq n$, berechne die Größe der Entenfamilie, zu der die Ente an Position (i,j) gehört, sofern $A[i,j] =$ `Ente`; ansonsten gib 0 zurück.

Begründen Sie die Korrektheit und geben Sie die Zeitkomplexität Ihres Algorithmus an!

Aufbauaufgaben

9. Aufgabe

1 + 1 + 2 Punkte

Beweisen Sie die folgenden nützlichen Lemmata:

a) Für alle $\alpha, \varepsilon \in \mathbb{R}^+$ gilt

$$n^\alpha \in o\big(n^{\alpha+\varepsilon}\big) \ . \qquad \text{(Lemma A: Poly-Hierarchie)}$$

b) Für alle $c, \varepsilon \in \mathbb{R}^+$ gilt

$$c^n \in o\big((c+\varepsilon)^n\big) \ . \qquad \text{(Lemma A: Exp-Hierarchie)}$$

c) Für alle $c \in \mathbb{R}^+$ gilt

$$\omega\left(c^n\right) \ni n! \in o\left(n^n\right) \ . \qquad \text{(Lemma A: Fakultät)}$$

10. Aufgabe

7 + 3 Punkte

Es sei eine $n \times n$-Matrix $A : [1:n] \times [1:n]$ gegeben, in der jedes Element $A[i,j]$ die Werte `Wasser` oder `Ente` annehmen kann. Zwei Enten gehören zu derselben Entenfamilie, wenn sie entweder in horizontal, vertikal oder diagonal benachbarten Feldern "schwimmen".

a) Wir wollen herausfinden, wie stark je zwei Enten verwandt sind. Den Verwandtschaftsgrad modellieren wir mit

- der kleinsten Zahl k an Schritten *nur* über Enten (von einem Entenfeld zu einem benachbarten, wie in 9), um von Ente (i_1, j_1) zu Ente (i_2, j_2) zu gelangen, *wenn die beiden zu einer Familie gehören*, und sonst
- der kleinsten Zahl k an Schritten über Wasser[1], um von einem Verwandten der Ente (i_1, j_1) zu einem Verwandten von Ente (i_2, j_2) zu gelangen. In diesem Fall wollen wir $-k$ zurückgeben.

Entwerfen Sie einen Algorithmus, der den Verwandtschaftsgrad für zwei gegebene Enten(indizes) berechnet!

Begründen Sie die Korrektheit und geben Sie die Zeitkomplexität Ihres Algorithmus an!

[1]Schritte über Enten sind erlaubt, zählen aber nicht. Intuition: es müssen noch (mindestens) k Enten zielsicher in die Welt gesetzt werden, um die Familien zu vereinen.

b) Martha möchte die Entenfamilie nicht mehr in ihrem Teich haben. Sie beauftragt Ihren Mann Kurt damit, der Plage ein Ende zu bereiten. Kurt hat Skrupel und möchte so wenige Enten wie möglich töten. Er recherchiert und findet heraus, dass eine Entenfamilie die Heimat wechselt, wenn man sie in (mindestens zwei) kleinere Familien zerschlägt.

   Entwerfen Sie einen Algorithmus für Kurt! Gegeben eine Teichmatrix `A` mit genau einer Entenfamilie soll dieser die Mindestanzahl Enten zurückgeben, die Kurt töten muss (und idealerweise deren Positionen).

   Begründen Sie die Korrektheit und geben Sie die Zeitkomplexität Ihres Algorithmus an!

Hinweis: In allen Algorithmenentwurfsaufgaben gilt: Stellen Sie zu Beginn die Kernidee(n) Ihres Algorithmus heraus; damit erleichtern Sie dem Korrektor das Verständnis ungemein. Der Algorithmus selbst sollte in *klarem, definiten* Pseudocode angegeben werden. Die Faustregel ist, dass ein Programmierer ohne Kenntnis der Problemstellung in der Lage sein sollte, den Algorithmus anhand Ihres Pseudocodes ohne Nachdenken in einer beliebigen Sprache zu implementieren. Insbesondere sollte Pseudocode von Sprachdetails abstrahieren und ohne spezielle Kenntnisse verständlich sein.

11. Aufgabe

4 + 3 Punkte

a) Beschreibe $A : [0 : m] \times [0 : m] \to \mathbb{Z}$ ein 2-dimensionales Feld mit den Elementen $A[i_1, i_2]$, $0 \leq i_1 + i_2 \leq m$.

   1. Bestimmen Sie eine sequentielle Speicherabbildungsfunktion

   $$\mathrm{LOC_A} : \{(i_1, i_2) \in [0 : m] \times [0 : m] \mid 0 \leq i_1 + i_2 \leq m\} \to [N : M] \ ,$$

   die in Zeit $\mathcal{O}(1)$ berechnet werden kann und sequentiell ist; beweisen Sie Ihre Behauptungen! Wie ist M zu wählen?

   2. Angenommen, man ließe $0 \leq i_1 + i_2 \leq 2m$ zu. Wie würde sich diese Erweiterung auf die Eigenschaften Ihrer Speicherabbildungsfunktion aus 1. auswirken?

b) Sei $A : [1 : m] \times \cdots \times [1 : m] \to \mathbb{Z}$ ein k-dimensionales Feld mit den Elementen $A[i_1, i_2, \ldots, i_k]$, $1 \leq i_k \leq i_{k-1} \leq \cdots \leq i_2 \leq i_1 \leq m$ (Verallgemeinerung der unteren Dreiecksmatrix).

   Bestimmen Sie eine lexikographische Speicherabbildungsfunktion

   $$\mathrm{LOC}_A : \{(i_1, \ldots, i_k) \in [1 : m] \times \cdots \times [1 : m] \mid 1 \leq i_k \leq \cdots \leq i_1 \leq m\} \to [N : M] \ ,$$

   die in Zeit $\mathcal{O}(km + k^2)$ berechnet werden kann. Begründen Sie Ihre Behauptungen! Wie ist M zu wählen?