Grundzüge DS & Alg (WS14/15)

Lösungsvorschlag zu Aufgabenblatt 1

Aufgabe 1

```
procedure IS(A)
    n = length (A)
    for i = 2 to n do
        x = A[i]
        j = i
        while j > 1 and A[j − 1] > x do
            A[j] = A[j − 1]
            j = j − 1
        A[j] = x
```

```
procedure QS(A)
    n = length (A)
    Quicksort (A, 1, n)
```

```
procedure Quicksort(A, i, j)
    if i < j then
        p = Partition (A, i, j, A[j])
        Quicksort (A, i, p − 1)
        Quicksort (A, p + 1, j)
```

```
procedure Partition(A, i, j, x)
    b = i − 1
    for k = i to j do
        swap (A[k], A[b + 1])
        if A[b + 1] ≤ x then
            b = b + 1
    return b
```

a)

$T_{IS}([23, 11, 8, 4, 15]) = 8$

$[23, 11, 8, 4, 15]$

nach 1. Iteration, 1 Vergleich: $[11, 23, 8, 4, 15]$

nach 2. Iteration, 2 Vergleiche: $[8, 11, 23, 4, 15]$

nach 3. Iteration, 3 Vergleiche: $[4, 8, 11, 23, 15]$

nach 4. Iteration, 2 Vergleiche: $[4, 8, 11, 15, 23]$

$T_{QS}([23, 11, 8, 4, 15]) = 10$

$[23, 11, 8, 4, 15]$

nach 1. Iteration, 5 Vergleiche: $[11, 8, 4]$ $[15]$ $[23]$

nach 2. Iteration, 3 Vergleiche: $[4]$ $[11, 8]$ $[15]$ $[23]$

nach 3. Iteration, 2 Vergleiche: $[4]$ $[8]$ $[11]$ $[15]$ $[23]$

b)

$A[] = [4, 5, 3, 1, 2]$

$T_{IS}([4, 5, 3, 1, 2]) = 10$

$[4, 5, 3, 1, 2]$

nach 1. Iteration, 1 Vergleich: $[4, 5, 3, 1, 2]$

nach 2. Iteration, 2 Vergleiche: $[3, 4, 5, 1, 2]$

nach 3. Iteration, 3 Vergleiche: $[1, 3, 4, 5, 2]$

nach 4. Iteration, 4 Vergleiche: $[1, 2, 3, 4, 5]$

$T_{QS}([4, 5, 3, 1, 2]) = 8$

$[4, 5, 3, 1, 2]$

nach 1, Iteration, 5 Vergleiche: $[1]$ $[2]$ $[5, 3, 4]$

nach 2. Iteration, 3 Vergleiche: $[1]$ $[2]$ $[3]$ $[4]$ $[5]$

c)

$A[] = [1, 2, 3, 4, 5]$

$T_{IS}([1, 2, 3, 4, 5]) = 4$

$[1, 2, 3, 4, 5]$

nach 1. Iteration, 1 Vergleich: $[1, 2, 3, 4, 5]$

nach 2. Iteration, 1 Vergleich: $[1, 2, 3, 4, 5]$

nach 3. Iteration, 1 Vergleich: $[1, 2, 3, 4, 5]$

nach 4. Iteration, 1 Vergleich: [1, 2, 3, 4, 5]

$T_{QS}([1, 2, 3, 4, 5]) = 14$

nach 1. Iteration, 5 Vergleiche: $[1, 2, 3, 4]$ $[5]$

nach 2. Iteration, 4 Vergleiche: $[1, 2, 3]$ $[4]$ $[5]$

nach 3. Iteration, 3 Vergleiche: $[1, 2]$ [3] [4] [5]

nach 4. Iteration, 2 Vergleiche: [1] [2] [3] [4] [5]

d)

$T_{IS}([5, 4, 3, 2, 1]) = 10$

Für jede Iteration i gilt:

$A[i]$ wird maximal mit allen vorherigen Schlüsseln $A[1, i-1]$ verglichen.

Außerdem gilt vor jeder Iteration i: $A[i] \leq A[j]$ für alle $j < i$.

Also wird bei jeder Iteration mit allen vorherigen Elementen verglichen und die Anzahl der Vergleiche ist damit maximal.

e)

$[1, 2, 3, 4, 5]$

Seien x und y zwei beliebige Elemente der zu sortierenden Liste. Sie können dann während der Durchfürung des Quicksort-Algorithmus maximal einmal miteinander verglichen werden. Nämlich in dem Fall, dass es sich bei x oder y um das Pivot-Element handelt, welches danach in keiner der weiter zu sortierenden Teillisten mehr auftritt.

Bei der aufsteigend sortierte Liste $[1, ..., n]$ wird jedes Paar von Elementen genau einmal miteinander verglichen. Deshalb können wir mit Hilfe der eben gefundenen oberen Schranke folgern, dass diese Liste die maximale Anzahl an Vergleichen benötigt. Ein Vergleich zwischen i und j $(i < j)$ findet in der $(n - j + 1)$-ten Iteration statt, wenn Partition() auf das Intervall $[1, j]$ aufgerufen wird und j das Pivot-Element ist.

f)

Für Insertion Sort: $A[]$ absteigend sortiert. Beweis: siehe d)

Für Quicksort: $A[]$ aufsteigend sortiert. Beweis: siehe e)

Aufgabe 2

(a)
Gegeben drei Funktionen f, g und h. Wir wissen $f \leq_{\text{fü}} g$ und $g \leq_{\text{fü}} h$. Zu zeigen ist, dann auch $f \leq_{\text{fü}} h$ gilt.

$f \leq_{\text{fü}} g = \exists n_0 \in \mathbb{N} : \forall n > n_0 : f(x) \leq g(x)$
$g \leq_{\text{fü}} h = \exists n_1 \in \mathbb{N} : \forall n > n_1 : g(x) \leq h(x)$

Daraus folgt direkt mit $n_2 = \max\{n_0; n_1\}$:

$\forall n > n_2 : f(x) \leq h(x) \Rightarrow f \leq_{\text{fü}} h$

(b)
Nein, denn es ist möglich, dass für zwei Funktionen f und g beispielsweise $f \nleq_{\text{fü}} g$ gilt, $f >_{\text{fü}} g$ aber nicht.

Bewiesen wird es durch folgendes Beispiel:

$$f(x) = \begin{cases} 1, & x \mod 2 = 0 \\ 0, & sonst \end{cases}$$

$$g(x) = \begin{cases} 0, & x \mod 2 = 0 \\ 1, & sonst \end{cases}$$

Wie man sieht gibt es unendlich viele Fälle in denen f größer als g ist (geraden Zahlen) und unendlich viele Fälle in denen g größer als f ist (ungeraden Zahlen). Somit gilt $f \nleq_{\text{fü}} g$ und $g \nleq_{\text{fü}} f$, $f >_{\text{fü}} g$ und $g >_{\text{fü}} f$ aber nicht.

Aufgabe 3

Ein Polynom vom Grad k lässt sich wie folgt schreiben:

$$f(n) = a_0 n^0 + a_1 n^1 + \cdots + a_k n^k$$

(a)
z.z. $f(n) = \Theta(n^k)$

$$\Leftrightarrow \exists c_1 > 0 : f(n) \leq_{\text{fü}} c_1 \cdot n^k \ \wedge \ \exists c_2 > 0 : c_2 \cdot n^k \leq_{\text{fü}} f(n)$$

Man muss zeigen, dass solche c_1 und c_2 existieren.

$$f(n) \leq_{\text{fü}} c_1 \cdot n^k$$

$$\Leftrightarrow a_0 n^0 + a_1 n^1 + \cdots + a_k n^k \leq_{\text{fü}} c_1 \cdot n^k$$

$$\Leftrightarrow \frac{a_0}{n^k} + \frac{a_1}{n^{k-1}} + \cdots + \frac{a_{k-1}}{n} \leq_{\text{fü}} c_1 - a_k$$

$$\Leftrightarrow \exists n_0 > 0 : \text{wenn } n > n_0,\ \frac{a_0}{n^k} + \frac{a_1}{n^{k-1}} + \cdots + \frac{a_{k-1}}{n} \leq c_1 - a_k.$$

Man wählt $n_0 = \max\{|a_0|, |a_1|, \ldots, |a_{k-1}|\}$, und $c_1 = a_k + k$, und erhält man die wahre Aussage:

$$\text{wenn } n > n_0,\ \frac{a_0}{n^k} + \frac{a_1}{n^{k-1}} + \cdots + \frac{a_{k-1}}{n} \leq 1 + 1 + \cdots + 1 = a_k + k - a_k.$$

Als $f(n)$ "füp" ist, folgt $a_k > 0 \Rightarrow c_1 > 0$.

$$c_2 \cdot n^k \leq_{\text{fü}} f(n)$$

$$\Leftrightarrow c_2 \cdot n^k \leq_{\text{fü}} a_0 n^0 + a_1 n^1 + \cdots + a_k n^k$$

$$\Leftrightarrow c_2 - a_k \leq_{\text{fü}} \frac{a_0}{n^k} + \frac{a_1}{n^{k-1}} + \cdots + \frac{a_{k-1}}{n}$$

$$\Leftrightarrow \exists n_0 > 0 : \text{wenn } n > n_0,\ c_2 - a_k \leq \frac{a_0}{n^k} + \frac{a_1}{n^{k-1}} + \cdots + \frac{a_{k-1}}{n}.$$

Man wählt $n_0 = \frac{2k}{a_k} \cdot \max\{|a_0|, |a_1|, \ldots, |a_{k-1}|, |a_k|\}$, und $c_2 = \frac{a_k}{2}$, und erhält man die wahre Aussage:

$$\text{wenn } n > n_0,\ -\frac{a_k}{2} = -\frac{a_k}{2k} - \frac{a_k}{2k} - \cdots - \frac{a_k}{2k} \leq \frac{a_0}{n^k} + \frac{a_1}{n^{k-1}} + \cdots + \frac{a_{k-1}}{n}.$$

Als $f(n)$ "füp" ist, folgt $a_k > 0 \Rightarrow c_2 > 0$.

$$\Rightarrow f(n) = \Theta(n^k) \quad q.e.d.$$

(b)
z.z. Für $l > k$ gilt $f(n) \in o(n^l)$

$$\Leftrightarrow \forall c > 0 : f(n) \leq_{\text{fü}} c \cdot n^l$$

Wenn man nun durch n^l teilt, erhält man:

$$\frac{a_0}{n^l} + \frac{a_1}{n^{l-1}} + \cdots + \frac{a_k}{n^{l-k}} \leq_{\text{fü}} c$$

Die obige Ungleichheit muss für $n \geq n_0$ gelten, für ein beliebiges aber festes n_0. Wenn $c < 1$, wählt man $n_0 = \frac{k+1}{c} \cdot \max\{|a_0|, |a_1|, \ldots, |a_{k-1}|\}$, und erhält man die wahre Aussage:

$$\frac{a_0}{n^l} + \frac{a_1}{n^{l-1}} + \cdots + \frac{a_k}{n^{l-k}} \leq (k+1)\frac{c}{k+1} = c.$$

Wenn $c \geq 1$, wählt man $n_0 = (k+1) \cdot \max\{|a_0|, |a_1|, \ldots, |a_{k-1}|\}$, und erhält man die wahre Aussage:

$$\frac{a_0}{n^l} + \frac{a_1}{n^{l-1}} + \cdots + \frac{a_k}{n^{l-k}} \leq (k+1)\frac{1}{k+1} = 1 \leq c. \quad q.e.d.$$

(c)
z.z. Für $l < k$ gilt $f(n) \in \omega(n^l)$

$$\Leftrightarrow \forall c > 0 : c \cdot n^l \leq f(n)$$

Man geht nun ähnlich vor wie bei Aufgabenteil (b).

Aufgabe 4

(a)

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$$

$$\Rightarrow \forall \varepsilon > 0, \exists n_\varepsilon : \forall n > n_\varepsilon, c - \varepsilon < \frac{f(n)}{g(n)} < c + \varepsilon$$

$$\Rightarrow \forall \varepsilon > 0 : (c-\varepsilon)g(n) <_{f\ddot{u}} f(n) <_{f\ddot{u}} (c+\varepsilon)g(n).$$

Man wählt ε so dass $\varepsilon < c$, und $c_1 = c + \varepsilon$, $c_2 = c - \varepsilon$.

$$\Rightarrow f(n) \leq_{f\ddot{u}} c_1 \cdot g(n), \text{ und } c_2 \cdot g(n) \leq_{f\ddot{u}} f(n)$$

$$\Rightarrow f(n) \in \Theta(g(n)).$$

(b)(c) Ähnlich wie bei Aufgabenteil (a).

Aufgabe 5

Im Folgenden wird für den asymptotischen Vergleich die Grenzwertbetrachtung aus 4b) verwendet. Die Transitivität wurde bereits in Aufgabe 2a) gezeigt. Es ergibt sich folgende Reihenfolge in Hinblick auf das asymptotische Wachstum:

$n^{1-\epsilon}$

$n/\log(n)$

$n/10$

$n \cdot \log(n)$

$n \cdot 2^{\sqrt{\log(n)}}$

$n^{5/3}$

$n^{5/3} \cdot \log(n)$

$(5/3)^n$

3^n

Beweise:

$n^{1-\epsilon} \in o(n/\log(n))$

$$\lim_{n\to\infty} \frac{n^{1-\epsilon}}{n/\log(n)} = \lim_{n\to\infty} \frac{\log(n)}{n^\epsilon}$$

Wir verwenden die Regel L'Hospital:

$$\lim_{n\to\infty} \frac{\frac{1}{n}}{\epsilon \cdot n^{\epsilon-1}} = \lim_{n\to\infty} \frac{1}{\epsilon \cdot n^\epsilon} = 0$$

$n/\log(n) \in o(n/10)$

$$\lim_{n\to\infty} \frac{n/\log(n)}{n/10} = \lim_{n\to\infty} \frac{10}{\log(n)} = 0$$

$n/10 \in o(n \cdot \log(n))$

$$\lim_{n\to\infty} \frac{n/10}{n \cdot \log(n)} = \lim_{n\to\infty} \frac{1}{10 \cdot \log(n)} = 0$$

$n \cdot \log(n) \in o(n \cdot 2^{\sqrt{\log(n)}})$

Mit $m = \sqrt{\log(n)}$:

$$\lim_{n\to\infty} \frac{n \cdot \log(n)}{n \cdot 2^{\sqrt{\log(n)}}} = \lim_{n\to\infty} \frac{\log(n)}{2^{\sqrt{\log(n)}}} = \lim_{m\to\infty} \frac{m^2}{2^m} = 0$$

$n \cdot 2^{\sqrt{\log(n)}} \in o(n^{5/3})$

Mit $m = \sqrt{\log(n)}$:

$$\lim_{n\to\infty} \frac{n \cdot 2^{\sqrt{\log(n)}}}{n^{5/3}} = \lim_{n\to\infty} \frac{2^{\sqrt{\log(n)}}}{n^{2/3}} = \lim_{n\to\infty} \frac{\sqrt{\log(n)}}{\log(n^{2/3})}$$

$$\lim_{n\to\infty} \frac{\sqrt{\log(n)}}{2/3 \cdot \log(n)} = \lim_{m\to\infty} \frac{m}{2/3 \cdot m^2} = \lim_{m\to\infty} \frac{3}{2m} = 0$$

$n^{5/3} \in o(n^{5/3} \cdot \log(n))$

$$\lim_{n\to\infty} \frac{n^{5/3}}{n^{5/3} \cdot \log(n)} = \lim_{n\to\infty} \frac{1}{\log(n)} = 0$$

$n^{5/3} \cdot \log(n) \in o((5/3)^n)$

$$\lim_{n\to\infty} \frac{n^{5/3} \cdot \log(n)}{(5/3)^n} = 0$$

$(5/3)^n \in o(3^n)$

$$\lim_{n\to\infty} \frac{(5/3)^n}{3^n} = \lim_{n\to\infty} (5/9)^n = 0$$