Sortieren:

(i) mit AVL-Bäumen

Array A, n Elemente

↓

AVL Baum aufbauen mit Werten aus A (Einfügen jeweils $O(\log n)$) — $O(n \log n)$

↓

AVL Baum in sortierter Reihenfolge ausgeben + nach A schreiben — $O(n)$

$O(n \log n)$

Untere Schranke für Sortieren (allgemein): $\Omega(n \log n)$
[siehe VL ~~nächste Woche~~]

Warum noch Mergesort? Speicheraufwand!

mit AVL-Baum: $O(n)$ zusätzlicher Speicher

~~bei~~ Mergesort: $O(1)$ —"—

Mergesort(A,p,r)

Input : Subarray von $A[1, ..., n]$, der bei p beginnt und bei r endet ($A[p, ..., r]$)
Output : Den sortierten Subarray

```
if (p < r) then
    q ← ⌊(p+r)/2⌋
    Mergesort(A, p, q)
    Mergesort(A, q + 1, r)
    Merge(A, p, q, r)
end
```

Merge(A,p,q,r):

```
n1 ← q − p + 1
n2 ← r − q
create arrays L[1, ..., n1 + 1] and R[1, ..., n2 + 1]
for (i ← 1 to n1) do
    L[i] ← A[p + i − 1]
end
for (j ← 1 to n2) do
    R[j] ← A[q + j]
end
L[n1 + 1] ← ∞
R[n2 + 1] ← ∞
i ← 1
j ← 1
for k ← p to r do
    if (L[i] ≤ R[j]) then
        A[k] ← L[i] i = i + 1
    end
    else
        A[k] ← R[j]
        j ← j + 1
    end
end
```

(ii) <u>Mergesort</u>

II

A: 2 8 6 5

2 8 | 6 5

2 | 8 | 6 5

2 8 | 6 | 5

2 8 | 5 6

2 5 6 8

A[1] = 2, A[2] = 8, A[3] = 6, A[4] = 5

Mergesort (A, 1, 4)

$q = \lfloor \frac{5}{2} \rfloor = 2$

Mergesort (A, 1, 2)

$q = \lfloor \frac{3}{2} \rfloor = 1$

Mergesort (A, 1, 1) ← nichts passiert

Mergesort (A, 2, 2) ← — " —

Merge (A, 1, 1, 2) → 2 8 (*) ← nicht, sondern (**)

Mergesort (A, 3, 4)

$q = \lfloor \frac{7}{2} \rfloor = 3$

Mergesort (A, 3, 3) ← nichts passiert

Mergesort (A, 4, 4) ← nichts passiert

Merge (A, 3, 3, 4) → 5 6 (**)

Merge (A, 1, 2, 4) → 2 5 6 7

(*):

Merge(A, 1, 1, 2)
$\uparrow \uparrow \uparrow$
p q r

$n_1 = q - p + 1 = 1 - 1 + 1 = 1$
$n_2 = r - q = 2 - 1 = 1$

$L[1,2]$ $R[1,2]$

$i=1$: $L[1] = A[1+1-1] = A[1] = 2$
$j=1$: $R[1] = A[1+1] = A[2] = 8$
$L[2] = \infty$
$R[2] = \infty$
$i=1$
$j=1$
$k=1$: $L[1] \leq R[1]$? ($2 \leq 8$? ja)
↳ $A[1] = L[1] = 2$
$i=2$
$k=2$: $L[2] \not\leq R[1]$ ($\infty \not\leq 8$)
↳ $A[2] = R[1] = 8$
end

Somit: $A[1]=2$, $A[2]=8$

(**)
Merge(A, 3, 3, 4)
$\uparrow \uparrow \uparrow$
p q r

$n_1 = q - p + 1 = 3 - 3 + 1 = 1$
$n_2 = r - q = 4 - 3 = 1$
$L[1,2]$, $R[1,2]$
$i=1$: $L[1] = A[3+1-1] = A[3] = 6$
$j=1$: $R[1] = A[3+1] = A[4] = 5$

$L[2]=\infty$
$R[2]=\infty$
$i=1$
$j=1$

$k=3$: $L[1] \nleq R[1]$

$\hookrightarrow A[3]=R[1]=5$
$j=2$

$k=4$: $L[1] \leq R[2]$ ✓ $(6 \leq \infty)$

$\hookrightarrow A[4]=L[1]=6$
$i=2$

Somit $A[3]=5$, $A[4]=6$

WS 08/09

V

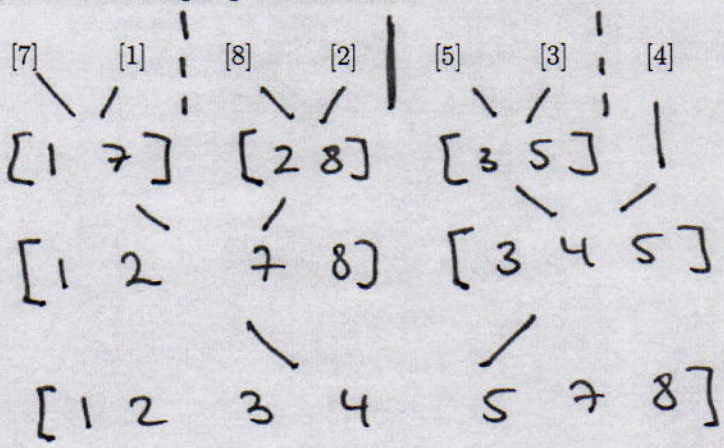
6.Aufgabe: Sortieren

9 Punkte

Sortiere die folgenden Zahlen mit dem in der Vorlesung vorgestellten Mergesort. Kennzeichne in jedem Schritt, welche Teilfolgen gemischt werden.

$p = 1,\ r = 7$

$q = \lfloor \frac{p+r}{2} \rfloor = 4$



VII

a) $T(n) = 256\, T\left(\frac{n}{4}\right) + n^3$

$$= \sum_{i=1}^{256} T\left(\frac{1}{4} \cdot n\right) + \Theta(n^3)$$

Also: $\alpha_i = \frac{1}{4}$, $i = 1, \dots, 256$

$m = 256$

$k = 3$

$$\sum_{i=1}^{m} \alpha_i^{k} = \sum_{i=1}^{256} \left(\frac{1}{4}\right)^3 = 256 \cdot \frac{1}{64} > 1$$

↳ 3. Fall, wir suchen also c mit $\sum_{i=1}^{m} \alpha_i^{c} = 1$

$$\sum_{i=1}^{256} \left(\frac{1}{4}\right)^c = 1$$

$$\Leftrightarrow \quad 256 \left(\frac{1}{4}\right)^c = 1$$

$$\Leftrightarrow \quad \left(\frac{1}{4}\right)^c = \frac{1}{256}$$

$$\Leftrightarrow \quad 4^c = 256$$

$$\Leftrightarrow \quad c = \log_4 256$$

$$\Leftrightarrow \quad c = 4$$

$$\Rightarrow \quad T(n) \in \Theta(n^c) = \Theta(n^4)$$

b) $T(n) = 27 \cdot T\left(\frac{n}{3}\right) + n^3$

$\alpha_i = \cancel{27}\,1/3, \quad i = 1, \dots, 27$

$m = 27$

$k = 3$

$$\sum_{i=1}^{m} \alpha_i^k = \sum_{i=1}^{27} \left(\frac{1}{3}\right)^3 = 27 \cdot \frac{1}{27} = 1 \qquad \rightarrow \text{Fall 2}$$

$$\Rightarrow T(n) \in \Theta(n^k \log n) = \Theta(n^3 \log n)$$

c) $T(n) = 3\, T\left(\frac{n}{4}\right) + n^2$

$\alpha_i = \frac{1}{4}, \quad i = 1, \dots, 3$

$m = 3$

$k = 2$

$$\sum_{i=1}^{m} \alpha_i^k = \sum_{i=1}^{3} \left(\frac{1}{4}\right)^2 = 3 \cdot \frac{1}{16} < 1 \qquad \rightarrow \text{Fall 1}$$

$$\Rightarrow T(n) \in \Theta(n^k) = \Theta(n^2)$$

O-Notation

↳ A1b) Ü3 – Kreuze an, in welchen Klassen die jeweilige Funktion liegt...

Erstmal: $f(n) \in O(1) \Rightarrow f(n) \in O(n)$ usw. !!

$f(n) \in O(1) \Leftrightarrow \exists c>0, n_0 \in \mathbb{N}: 0 \leq f(n) \leq c \cdot 1 \quad \forall n \geq n_0$

Was brauchen wir für $f(n) \in O(n)$

$\exists c_2>0, n_{0_2} \in \mathbb{N}: 0 \leq f(n) \leq c_2 \cdot n \quad \forall n \geq n_{0_2}$

Da $c \cdot 1 \leq c_2 \cdot n$ für geeignetes c_2 (z.B. $c_2 = c$) gilt dies.

Klassen: $O(1), O(n), O(2^n), \Omega(1), \Omega(n), \Omega(2^n),$
$\Theta(1), \Theta(n), \Theta(2^n)$

- $f(n) = 57$? $O(1), O(n), O(2^n), \Omega(1), \Theta(1)$
- $f(n) = 3n - 51$? $O(n), O(2^n)$
  $\Omega(1), \Omega(n)$
  $\Theta(n)$
- $f(n) = n^2$? $O(2^n)$
  $\Omega(1), \Omega(n)$
- $f(n) = n^n$? $\Omega(1), \Omega(n), \Omega(2^n)$
- $f(n) = n^3$? $O(2^n)$
  $\Omega(1), \Omega(n)$

...