Tutorium 09-06-24

Notiztitel 5/6/2009

Deterministisch vs Nicht deterministisch

Θ-Notation

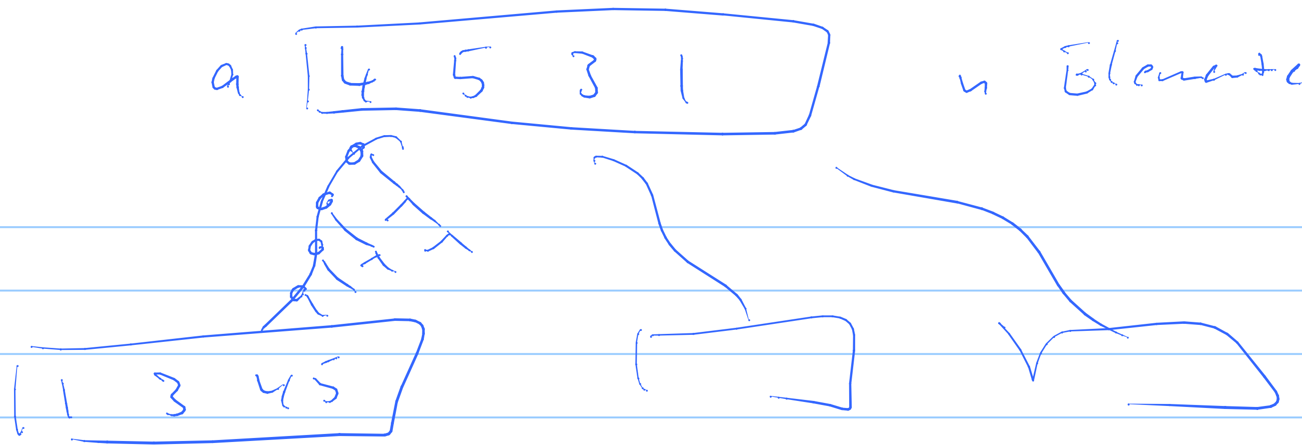
Sortieren mindestens $\Theta(n \cdot \log n)$?

→ 8.6 b, 8.7

8.5, P NP und ähnliches ← Berechenbarkeit

Sortieren: "Wenn Vergleiche nötig sind, dann braucht jeder Sortieralgorithmus $n \cdot \log_2 n$ Schritte im worst-case"

Wenn Vergleiche nötig sind, muß beim Sortieren per Vergleich fest gestellt werden, wie die neue Reihenfolge aussieht



Umordnung der Ausgangsliste

Es gibt $n!$ Umordnungen einer Liste

Sortierergebnis ist eine dieser Umordnungen

Welche davon hängt nur an Originalliste, nicht am Algor.

Um festzustellen, ob etwas getauscht werden muß, muß ich vergleichen a_i mit a_j

- Mit wieviel Vergleichen kann ich sicher sein, daß die Reihenfolge, die entsteht, richtig ist?

d.h. die Tiefe des Vergleichsbaums bestimmt Laufzeit

- Ein Vergleich fächert Vergleichsbaum in 2 Möglichkeiten auf

mit k Vergleichen kann ich maximal 2^k verschiedene Listen erzeugen

- Für welches k ist $2^k \geq n!$

Laufzeit im worst case ist dieses k

$u \geq n \cdot \log n$

$2^u \geq 2^{n \cdot \log n}$

$\underline{\Theta(n^2)} \Rightarrow$ Problem $\in \underline{\underline{O(n^4)}}$

average case

Bücher $\rightarrow$ $\boxed{\begin{array}{l}\text{Komplexität von Algorithmen} \\ \text{Approximierende Algorithmen} \\ \text{Randomisierte} \quad \text{"}\end{array}}$

<u>8.6b</u> $\quad 2^{\sqrt{n}} \quad / \quad n^{\log n}$ $\quad\quad \log_2 n$

<u>zeige</u> $\quad n^k < \underset{n^{\log n}}{2^{\sqrt{n}}} < c^n$ $\quad\quad$ = die Zahl x mit $2^x = n$

$(k\cdot\log_2 10)^4$

$\sqrt{n}$ wächst stärker als $k\cdot\log n$ für große n

$$\boxed{n^k = \left(2^{\log n}\right)^k = 2^{\boxed{k\cdot\log n}} < 2^{\sqrt{n}}}$$

wähle $n \geq k^4$ $\quad \sqrt{n} = k^2 > k\cdot\log k$

für alle $n > k^4$ ist $n^k < 2^{\sqrt{n}}$

$\log n := \log_{10} n \qquad n = 10^{\log n} = 2^{\log_2 10\cdot\log n}$

3

$c>1$ $\quad \boxed{c^n = 2^{\log_2 c\cdot n} > 2^{\sqrt{n}}}$

für $n > \left(1/\log_2 c\right)^2$

$\sqrt{n}$ wächst schwächer als n

$n^k = 2^{\boxed{k\cdot\log n}}$ $\quad <$

$n^{\log n} = \left(2^{\log n}\right)^{\log n} = 2^{\boxed{(\log n)^2}}$ $\quad <$

$c^n = 2^{\boxed{n\cdot\log_2 c}}$

„naiv $O(2^n)$"

Faktorisierung / „diskrete Logarithmen"
Kryptographie in $O(2^{\sqrt{n}})$ $\quad n$ = Zahl der bits

8.7 $\max(f,g)(n) \overset{\in \mathbb{R}^+}{=} \max\{f(n), g(n)\}$

$\hookrightarrow \in \Theta(f+g)$

$\max(f,g) \leq_a c_1 \cdot (f+g)$

$\geq_a c_2 (f+g)$

für alle n $\max(f,g)(n) = \max\{f(n), g(n)\}$

$\leq f(n) + g(n)$ weil beide positiv

für alle n $\underline{f(n)+g(n)} \leq 2 \cdot \max\{f(n), g(n)\}$

wähle $c_1 = 1$, $c_2 = \frac{1}{2}$

dann für alle n $c_2 \cdot (f+g)(n) \leq \max(f,g)(n)$

$\leq c_1 \cdot (f+g)(n)$

8.7b $f \cdot g \in O(f) \iff g \in O(1)$

$\exists\, n_0 \in \mathbb{N}$ sodaß . . . .

$g \in O(1) \iff g(n) \le c$ für alle $n \ge n_0$

$f \cdot g \in O(f) \iff f(n) \cdot g(n) \le c' \cdot f(n)$ für alle $n \ge n_1$

„$\Leftarrow$" Sei $g \in O(1)$ dann

$g(n) \le c$ für alle $n \ge n_0$

also $f(n) \cdot g(n) \le c \cdot f(n)$ " " "

„$\Rightarrow$" also $g(n) \le c \cdot (f(n)/f(n))$ " "

weil $f(n) > 0$

also $g(n) \le c$ " "

8.5 „Typische Hausaufgabe"

$PKP = \{ (u_1, v_1) \in \Sigma^+ \mid \exists\, i_1 \ldots i_n \; . \; u_{i_1} \ldots u_{i_n} = v_{i_1} \ldots v_{i_n} \}$

$MPKP = \{ (\quad) \mid \exists i_1..i_n .\ u_1 u_{i_1} .. u_{i_n} = v_1 v_{i_1} .. v_{i_n} \}$

$M'PKP = \{ (u_i, v_i) \in \Sigma^*, w \in \Sigma^+ \mid \exists i_1..i_n .\ w u_{i_1} .. u_{i_n} = v_{i_1} .. v_{i_n} \}$

zu zeigen $PKP \leq M'PKP \leq MPKP$

(es gibt f, g total bb mit $x \in PKP \Leftrightarrow f(x) \in M'PKP$

$y \in M'PKP \Leftrightarrow g(y) \in MPKP$)

in VL $H \leq MPKP \leq PKP$ "anwenden"

② $M'PKP \leq MPKP$ konstruiere neue Paare

w muß links im ersten Paar stehen

$(u_i, v_i) .. w \in M'PKP \overset{\text{Def } M'PKP}{\Longleftrightarrow} w u_{i_1} \; u_{i_n} = v_{i_1} .. v_{i_n}$

$\underset{a \text{ neu}}{\Longleftrightarrow} a w u_{i_1} .. u_{i_n} = a v_{i_1} .. v_{i_n}$

$\Updownarrow$ Def MPKP

$g((u_1, v_1) ... w) \in MPKP \Leftrightarrow \overbrace{(aw) u_{i_1} .. u_{i_n} = (a)}^{\text{erstes Paar}} v_{i_1} .. v_{i_n}$

$g((u_i, v_i) .. w) = (aw, a) (u_1, v_1) .. (u_n, v_n)$

a neu

Def PCP

(1) $(u_i, v_i) \in PCP \iff \exists i_1 \ldots i_m : u_{i_1} \ldots u_{i_m} = v_{i_1} \ldots v_{i_m}$

$\iff \boxed{a}\, u_{i_1} \ldots u_{i_m} = \boxed{(a v_{i_1})} \ldots v_{i_m}$

wähle $i_1' = n + i_1$

$i_j' = i_j$ sonst

$f(u_i, v_i) \in MPCP \iff$ " $a \cancel{\#}\, u_{i_1'} \ldots u_{i_m'} = v_{i_1'} \ldots v_{i_m'}$

brauche nur ein Wort w — wähle neues a

$f((u_i, v_i)) = (u_i, v_i)_1^n, (u_i, a v_i)_1^n, a$

Idee einfach ~~~~ genauer Nachweis formal mühsam 'verbal einfach'!

Probeklausur im Web nächste Woche

Projekte "anmelden" lassen