

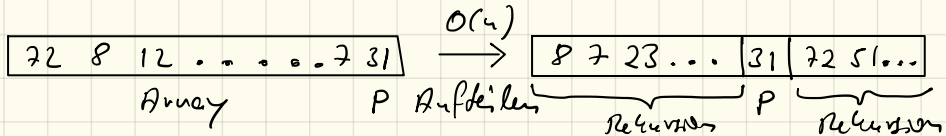
Algorithmen & Datenstrukturen

Herbst 2020

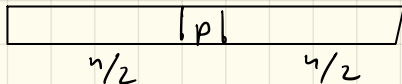
Vorlesung 15

Quicksort rekursiv:

Sortiere Array mit n Zahlen



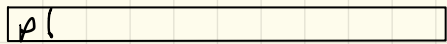
Ideal: p ist Median



← eine Hälfte kann ein
kleiner
sein, aber
egal

$$T(n) = 2T(n/2) + c \cdot n \Rightarrow O(n \log n)$$

Worst case: 2.B. inner



$$T(n) = T(n-1) + c \cdot n \Rightarrow O(n^2)$$

sehr unwahrscheinlich $\Rightarrow$ average case $O(n \log n)$
(wenn zufälliges p) (Analyse: nächstes Semester)

Frage: Wie teuer wäre es den Median zu berechnen? z.B. Median in $O(n)$
 $\Rightarrow$ Quicksort immer $O(n \log n)$

Auswahlproblem (Selection problem)

gegeben: Array $A[1], \dots, A[n]$

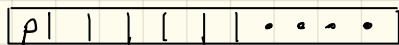
gesucht: i -kleinstes Element Spezialfall: Median
= $\lceil n/2 \rceil$ kleinstes

Idee 1: i mal Minimum entfernen $\Theta(in)$

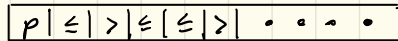
Idee 2: Sortieren, i -tes ausgeben $\Theta(n \log n)$

Idee 3: Pivotalieren, rekursiv

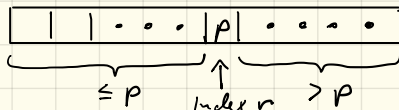
$\Theta(1)$ 1.) wähle Pivot
(z.B. erstes)



$\Theta(n)$ 2.) zähle Elemente $\leq p$
($\Rightarrow$ r viele)



$\Theta(n)$ 3.) teile A



4.) $i = r$: return p

$i < r$: suche i -tes links

$i > r$: suche $(i-r)$ -tes rechts

heißt "quicksselect": wie quicksort aber Rekursion nur auf einer Hälfte

Laufzeit: worst case $T(n) = T(n-1) + \Theta(n)$

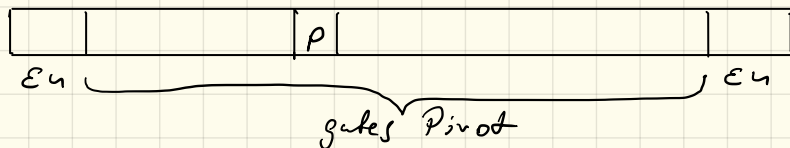
$$\Rightarrow T(n) = \Theta(n^2)$$

aber average case $T(n) = \Theta(n)$ (ohne Beweis)

Was ist ein gutes Pivotelement?

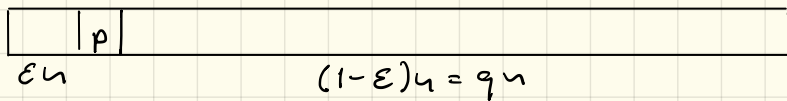
Antwort: reduziert Länge des Arrays in jeder Iteration mindestens um Faktor $q < 1$

Bild: Nach Aufteilen:



$0 < \epsilon < \frac{1}{2}$
fix!
z.B. $\epsilon = 0.1$

also schlimmstenfalls:



Denn dann: $T(n) \leq T(qn) + c \cdot n$, c konstant
($q = 1 - \epsilon < 1$)

teleskopieren: $\leq T(q^2 n) + cq + cn$

• • • • ($\log_{1/q}(n)$ Schritte)

$\leq T(1) + n \cdot c \cdot$ geometrische
Reihe in q

$= \mathcal{O}(n)$

Wie schaffen wir das deterministisch?
(d.h. immer, nicht nur im average case)

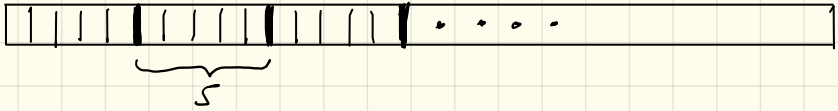
Algorithmus "Median der Mediane"

(Blum, Floyd, Pratt, Rivest, Tarjan 1973)

Idee: clevere Pivotwahl (neuer Schritt 1.)

1.) wähle Pivot wie folgt:

0(1) a.) Betrachte Array in Fünfergruppen:



0(1) b.) Bestimme Median in jeder Gruppe

0(1) c.) $A' =$ Array der Gruppenmediane (Länge $\lceil \frac{n}{5} \rceil$)

d.) Bestimme Median von A' : Median = Pivot p

wie? Rekursion: $\lceil \frac{1}{2} \lceil \frac{n}{5} \rceil \rceil$ -tes Element von A'

2.) - 4.) wie vorher (= Aufteilen, Rekursion links oder rechts)

Also zwei rekursive Aufrufe!

p ist Median der Gruppenmediane

Wie gut ist p ? (= wieviele Elemente in A sind garantiert kleiner und größer?)
= was ist ϵ von voriger Seite?)

Bild: wir wählen A so: (A ist nicht so sortiert)
 garantiert $\leq p \approx \frac{3}{10} n$ viele

jede Gruppe
sortiert

2	1	20	7	4		
3	7	27	51	5		
5	13	28	...	...	...	...
100	33	42	74	92	108	121
105	80	101	112	33		

$p = 73$

$\leftarrow A'$

sortiert nach A'

garantiert $\geq p \approx \frac{3}{10} n$ viele

Laufzeit: $T(n) \leq T(\frac{n}{5}) + T(\frac{7}{10}n) + cn$
 (Skizze)

$n - \frac{3}{10}n$

Zeige: $T(n) \leq an$ per Induktion, a konstant

Anfang: $T(n) \leq a \cdot n \quad \forall n \leq n_0$ (z.B. $n_0 = 100$)
 wähle a so daß das gilt und $a > 10c$

Schritt: Annahme $T(k) \leq ak \quad \forall k < n$

$$\Rightarrow T(n) \leq a \frac{n}{5} + a \frac{7}{10} n + cn$$

$$= a \cdot \frac{9}{10} n + cn$$

$$\leq an$$

$\square$

Insbesondere: Rekurs geht in $O(n)$

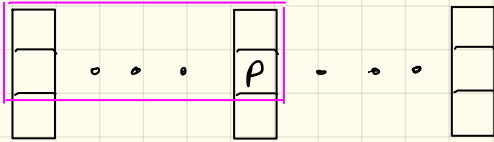
In der Praxis: zufälliges Pivot funktioniert gut

Warum Fünfergruppen?

$$T(n) \leq T\left(\frac{n}{5}\right) + T\left(\frac{7}{10}n\right) + cn$$

Summe = $\frac{9}{10}n < n$ Wichtig im Beweis!

Dreiergruppen: $\approx \frac{1}{3}n$



Rekurrenz $T(n) \leq T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + cn$

Summe = n

Viererguppen: $T(n) \leq T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + cn$

Ab $n \geq 5$ klappt es, n konstant!

$n/5$ - Gruppen: $T(n) \leq 5T\left(\frac{n}{5}\right) + T\left(\frac{7}{10}n\right) + cn$

zu teuer!

