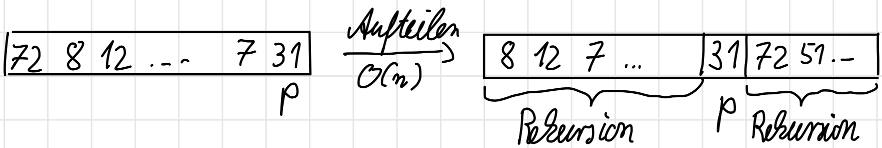
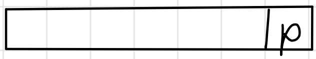
AUSWAHLPROBLEM, MEDIAN

Wiederholung QuickSort:

Sortiere $A[1 \ldots n]$

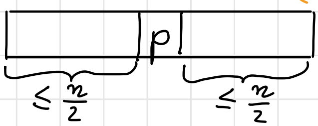


falls worst case immer auftritt

Worst case: z. B. 

$\rightsquigarrow T(n) = T(n-1) + cn \quad \rightsquigarrow T(n) = \Theta(n^2)$

mittleres Element   nicht verwechseln mit Mittelwert (mean, average $= \frac{1}{n}\sum_i A[i]$)

Best case: p ist Median 

$\rightsquigarrow T(n) \leq 2 \cdot T\left(\frac{n}{2}\right) + cn \quad \rightsquigarrow T(n) = O(n \log n)$

Good case, nächstes Semester 

, ε konstant, z. B. $\varepsilon = 0.1$

$T(n) = ??$

nächstes Semester: $T(n) = O(n \log n)$
Grund dafür, warum randomisiertes QuickSort schnell ist.

Idee: QuickSort läuft immer (deterministisch) in $O(n \log n)$,

wenn wir den Median in Zeit $O(n)$ berechnen können

Auswahlproblem

Input: $A[1 \dots n]$, $i \in \{1, \dots, n\}$

gesucht: i-kleinstes Element   Spezialfall: Median für $i = \frac{n+1}{2}$

Idee 1: Sortieren $\Theta(n \log n)$

Idee 2: Suchbaum $\Theta(n \log n)$ für Erzeugen des Suchbaums

optionales Material

Idee 3: Min-Heap erzeugen in $O(n)$, dann i-mal ExtractMin

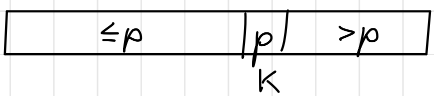
$\rightsquigarrow O(n + i \cdot \log n)$ auch $O(n \log n)$ für Median

Ziel: $O(n)$

Idee 4: Pivotieren, rekursiv:

QuickSelect (A, i)

1) Wähle Pivot p
2) $k = $ Aufteilen $(A[1..n], i)$ $O(n)$



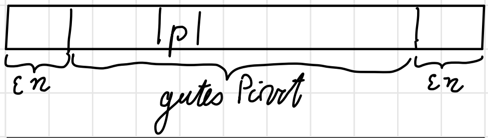
Wie QuickSort

3) $i = k$: return p
   $i < k$: QuickSelect $(A[1 \dots k-1], i)$
   $i > k$: QuickSelect $(A[k+1, n], i-k)$

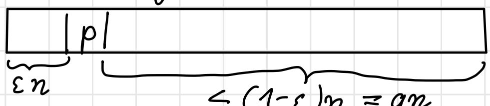
Wie wählen wir p?

Beliebig $\rightsquigarrow$ worst-case $T(n) = T(n-1) + cn \rightsquigarrow T(n) = \Theta(n^2)$

Gutes Pivot



worst-case



$, q = 1-\varepsilon < 1$

unter den guten Pivots

$\rightarrow T(n) \leq T(qn) + cn \quad , c$ konstant

teleskopieren: $\leq T(q^2 n) + c \cdot qn + cn$

$$\leq T(q^3 n) + c \cdot q^2 n + c \cdot qn + cn$$

$$\vdots$$

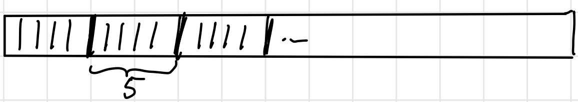
$$\leq T(1) + c \cdot n \cdot \underbrace{\sum_{i=0}^{\dots} q^i}_{\leq O(1)} \leq O(n)$$

Wie schaffen wir das immer? (Ohne auf Zufall zu vertrauen)

Algorithmus „Median der Mediane" (Blum, Floyd, Pratt, Rivest, Tarjan)

QuickSelect (A, i) mit neuem Schritt 1 (Pivotwahl):

$O(1)$ 1a) Betrachte A in Fünfergruppen:



$O(n)$ 1b) Bestimme Median jeder Gruppe

$A' =$ Array der Mediane (Länge $\lceil \frac{n}{5} \rceil$)

1c) $p =$ Median von A' mit QuickSelect $(A', \frac{1 + \lceil n/5 \rceil}{2})$

Rekursion

$O(n)$ 2) $k =$ Aufteilen $(A[1..n], p)$

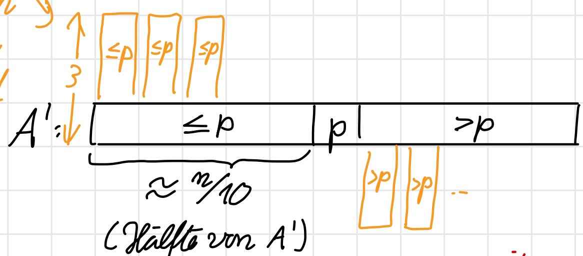
3) Rekursion

wie vorher

doppelte Rekursion!

Wie gut ist p? Wie viele Elemente sind kleiner als p?

jeder Median hat zwei Elemente in der Gruppe, die noch kleiner sind



insgesamt $\approx \frac{3}{10}n$ Elemente, die garantiert kleiner (größer) als p sind.

Wir ignorieren hier Ab-/Aufrunden. Ändert nichts am Ergebnis.

Laufzeit:

$$T(n) \leq \overbrace{T\left(\tfrac{n}{5}\right)}^{1c)} + \overbrace{T\left(\underbrace{\tfrac{7}{10}n}_{n-\frac{3}{10}n}\right)}^{3)} + \overbrace{c \cdot n}^{\text{Rest}}$$

$$T(1) \leq c$$

für eine geeignete Konstante c

Behauptung: $T(n) \leq 10 \cdot c \cdot n$

Beweis (Induktion): $n = 1$ ✓ (Induktionsanfang durch Rundungen eigentlich etwas komplizierter)

$n > 1$:
$$T(n) \leq T\left(\tfrac{n}{5}\right) + T\left(\tfrac{7}{10} \cdot n\right) + cn$$
$$\overset{\text{I.-H.}}{\leq} 10 \cdot c \cdot \tfrac{n}{5} + 10 \cdot c \cdot \tfrac{7}{10} n + c \cdot n$$
$$= 10c \cdot \left(\tfrac{9}{10} \cdot n\right) + cn$$
$$= 10cn$$

Entscheidend: $\frac{1}{5} + \frac{7}{10} = \frac{9}{10} < 1$ erlaubt es uns, die $+cn$ zu absorbieren. Würde für Dreiergruppen nicht aufgehen.

Also: Laufzeit $O(n)$!