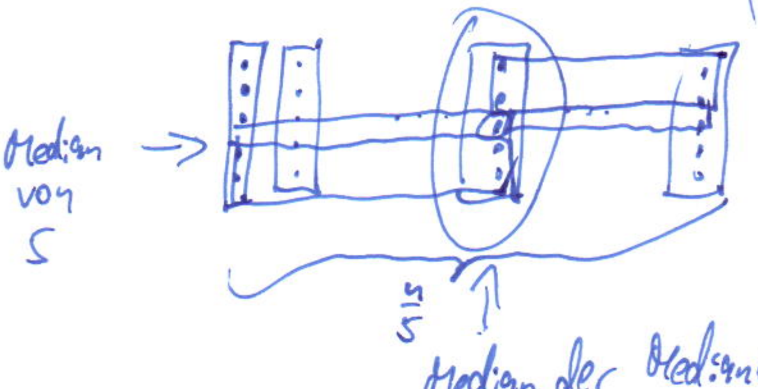
ALGORITHMUS 2.x (Median der Mediane)

Umsetzung: "SELECT" (Kapitel 9.3 im Cormen)

Eingabe: n Zahlen $x_1, \dots, x_n$, k

Ausgabe: Zahl vom Rang k

(Spezialfall: $k = \lceil \frac{n}{2} \rceil$: Median)



(1) Teile n Elemente in $\lfloor \frac{n}{5} \rfloor$ Fünfergruppen auf. (Eine kleinere Gruppe für den Rest) } $O(n)$

(2) Bestimme Mediane der Gruppen } $O(n)$

(3) Wende SELECT rekursiv an, für $\frac{n}{5}$ Mediane.

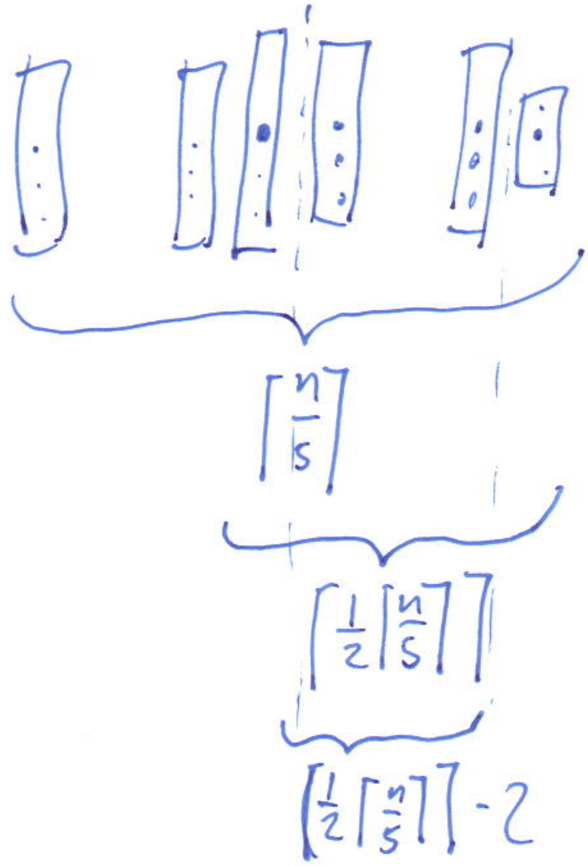
(4) Verwende Median der Mediane x als Pivot.
Das liefert $i-1$ kleinere, $n-i$ größere

(5) Falls $i = k$: x
Falls $i < k$: i-tes Element links
Falls $i > k$: $(i-k)$-tes Element rechts

wieviele Elemente sind kleiner bzw. größer?

Präzisierungen:

- Betrachten wir zur Vereinfachung nur verschiedene Elemente.
  (Bei Gleichheit werden die Mengen nur größer.)



Also:

- Mindestens $3\left(\left\lceil \frac{1}{2} \left\lceil \frac{n}{5} \right\rceil \right\rceil - 2\right) \geq \frac{3n}{10} - 6$ Elemente sind größer bzw. kleiner als der Median der Mediane
- Höchstens $\frac{7n}{10} + 6$ sind größer bzw. kleiner.

Laufzeitrekursion:

- Schritte (1), (2), (4) benötigen $O(n)$
- Schritt (3) benötigt $T(\lceil \frac{n}{5} \rceil)$
- Schritt (5) benötigt höchstens $T(\frac{7n}{10} + 6)$
  (denn $T(n)$ ist monoton wachsend)

Jetzt: Für $n \leq 140$ Laufzeit $O(1)$ ← „Magie" aber klar

Damit

$$T(n) \leq \begin{cases} O(1) & \text{falls } n < 100 \\ T(\lceil \frac{n}{5} \rceil) + T(\frac{7n}{10} + 6) + O(n) & \end{cases}$$

Mastertheorem:
$$T(a \cdot n) + T(bn) + O(n)$$
mit $a + b < 1$

Für $n \geq 100$: $\lceil \frac{n}{5} \rceil \leq \frac{n}{5} + 1 \leq \frac{2n+10}{10} \leq \frac{2{,}1n}{10}$

$$\frac{7n}{10} + 6 \leq \frac{7n+60}{10} \leq \frac{7{,}6n}{10}$$

$\frac{2{,}1 + 7{,}6}{10} = \frac{9{,}7}{10} < 1$

SATZ 2.(x+1) Blum, Floyd, Pratt, Rivest, Tarjan (BFPRT 1973)
RSA

Median der Mediane berechnet ein Element von Rang k in $O(n)$

2.3 Untere Schranken

Gesehen:
- Viele Standortprobleme in $O(n)$ lösbar
- Sortieren nicht besser als in $\Omega(n \log n)$ lösbar, wenn nur verglichen werden darf.

↳ „Entscheidungsbaum"-Argument aus A+D.

(Menge der möglichen Permutationen, jede Frage liefert bestenfalls Halbierung, $\log_2 n! \in \Theta(n \log n)$.)

Andere Probleme:

Problem 3.(x+2) (Element Distinctness)

Gegeben: n Zahlen, $x_1, \dots, x_n$

Frage: Sind alle verschieden oder gibt es zwei gleiche?

Problem 3.(x+3) (Max Gap)

Gegeben: n Zahlen, $x_1, \dots, x_n$, δ

Frage: Gibt es ~~zwei Zahlen~~ eine Lücke von mindestens Größe δ in der Menge $\{x_1, \dots, x_n\}$, d.h. zwei in der sortierten Folge

$$x_{\pi(1)} \leq x_{\pi(2)} \leq \dots \leq x_{\pi(n)}$$

aufeinanderfolgende Zahlen

$$x_{\pi(i-1)} \text{ und } x_{\pi(i)}$$

mit

$$x_{\pi(i-1)} + \delta \leq x_{\pi(i)}$$

Rechenmodell: „Algebraische ~~Vergl~~ Berechnungen": Wir dürfen einzelne Zahlen manipulieren und paarweise vergleichen, aber nicht globalere Sachen wie z.B. Hashing etc.

Verallgemeinerung von unterer Schranke für Vergleiche:
BEN-OR (1983) "Algebraic Computation Trees"

Damit:

SATZ 3.(x+4) (Lee + Wu 1986)
Untere Schranken von $\Omega(n \log n)$ für Element Distinctness und Max Gap im Algebraic-Computation-Tree-Modell