

# Kapitel III Selektieren und Sortieren

## 1. Einleitung

**Gegeben:** Menge  $S$  von  $n$  Elementen aus einem total geordneten Universum  $U$ ,  $i \in \mathbb{N}$ ,  $1 \leq i \leq n$ .

**Gesucht:**  $i$ -kleinstes Element in  $S$ .

Die Fälle  $i = 1$  bzw.  $i = n$  entsprechen der Suche nach dem Minimum bzw. Maximum.

Der Standardalgorithmus dafür benötigt  $n - 1$  Vergleiche.

## Satz 76

*Die Bestimmung des Minimums/Maximums von  $n$  Elementen benötigt mindestens  $n - 1$  Vergleiche.*

### Beweis:

Interpretiere Algorithmus als Turnier. Ein Spiel wird jeweils vom kleineren Element gewonnen. Wir beobachten: Jedes Element außer dem Gesamtsieger muss mindestens ein Spiel verloren haben  $\Rightarrow n - 1$  Vergleiche notwendig. □

## Bestimmung des Vize-Meisters bzw. des zweitkleinsten Elements

### Satz 77

Das zweitkleinste von  $n$  Elementen kann mit

$$n + \lceil \log_2 n \rceil - 2$$

Vergleichen bestimmt werden.

### Beweis:

Wir betrachten wiederum ein KO-Turnier:  $(n - 1)$  Vergleiche genügen zur Bestimmung des Siegers (Minimum).

Das **zweitkleinste** Element ist unter den „Verlierern“ gegen das Minimum zu suchen. Deren Anzahl ist  $\leq \lceil \log_2 n \rceil$ . Man bestimme nun unter diesen Elementen wiederum das Minimum und erhält damit das zweitkleinste Element in  $\leq \lceil \log_2 n \rceil - 1$  weiteren Vergleichen. □



Lewis Carroll:

*Lawn Tennis Tournaments*

St. Jones Gazette (Aug. 1, 1883), pp. 5–6

Reprinted in *The Complete Work of Lewis Carroll*. Modern Library, New York (1947)



Vaughan R. Pratt, Frances F. Yao:

*On lower bounds for computing the  $i$ -th largest element*

Proc. 14th Ann. IEEE SWAT, pp. 70–81 (1973)



Donald E. Knuth:

The art of computer programming. Vol. 3: Sorting and searching,

3. Auflage, Addison-Wesley Publishing Company: Reading (MA), 1997

## 2. Der Blum-Floyd-Pratt-Rivest-Tarjan Selektions-Algorithmus

### Definition 78

Sei  $n \in \mathbb{N}$ . Der **Median** (das „mittlere“ Element) einer total geordneten Menge von  $n$  Elementen ist deren  $i$ -kleinstes Element, wobei

$$i = \left\lceil \frac{n}{2} \right\rceil .$$

**Bemerkung:** Für gerade  $n$  wird manchmal auch  $i = \left\lfloor \frac{n}{2} \right\rfloor + 1$  benutzt.

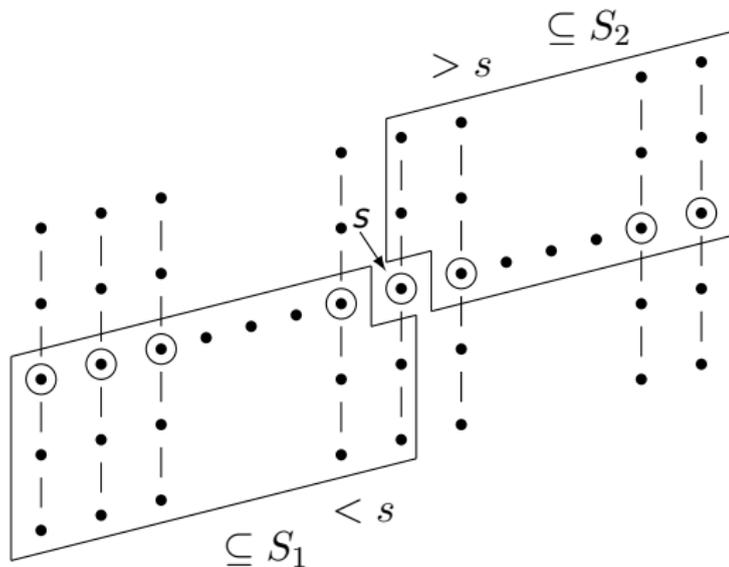
Sei  $m$  eine kleine ungerade Zahl (etwa  $5 \leq m \leq 21$ ). Sei  $S := \{a_1, \dots, a_n\}$  eine Menge von  $n$  paarweise verschiedenen Elementen. Zur Bestimmung des  $i$ -kleinsten Element in  $S$  betrachten wir folgenden Algorithmus BFPRT.

# Der BFPRT-Selektions-Algorithmus (1/3)

- 1 Teile  $S$  in  $\lceil \frac{n}{m} \rceil$  Blöcke auf,  $\lfloor \frac{n}{m} \rfloor$  davon mit je  $m$  Elementen
- 2 Sortiere jeden dieser Blöcke
- 3 Sei  $S'$  die Menge der  $\lceil \frac{n}{m} \rceil$  Mediane der Blöcke. Bestimme rekursiv den **Median**  $s$  dieser Mediane (also das  $\lceil \frac{|S'|}{2} \rceil$ -kleinste Element von  $S'$ ).

## Der BFPRT-Selektions-Algorithmus (2/3)

- ④ Partitioniere  $S - \{s\}$  in  $S_1 := \{x \in S : x < s\}$ ,  
 $S_2 := \{x \in S : x > s\}$ . Bemerkung:  $|S_1|, |S_2| \geq \frac{n}{4}$ , falls  
 $n \geq 3m - 1$ .



## Der BFPRT-Selektions-Algorithmus (3/3)

- 5 Falls  $i \leq |S_1|$ , bestimme rekursiv das  $i$ -kleinste Element in  $S_1$ .  
Falls  $i = |S_1| + 1$ , gib  $s$  als Lösung zurück.  
Ansonsten bestimme rekursiv das  $(i - |S_1| - 1)$ -kleinste Element in  $S_2$ .

Sei  $T(n)$  die worst-case Anzahl von Vergleichen für  $|S| = n$  des Algorithmus BFPRT. Sei  $C_m$  die # von Vergleichen, um  $m$  Elemente zu sortieren (z.B.  $C_5 = 7$ ,  $C_{11} = 26$ ). Es gilt:

$$T(n) \leq \underbrace{T\left(\left\lceil \frac{n}{m} \right\rceil\right)}_{3.} + \underbrace{T\left(\left\lfloor \frac{3}{4}n \right\rfloor\right)}_{5.} + \underbrace{\left\lceil \frac{n}{m} \right\rceil}_{2.} C_m + \underbrace{\left\lfloor \frac{n}{2} \right\rfloor}_{4.}$$

## Satz 79

*Der Selektions-Algorithmus BFPRT bestimmt das  $i$ -kleinste Element von  $n$  Elementen mit  $\mathcal{O}(n)$  Vergleichen (und Zeit).*

## Beweis:

Annahme:  $T(n) \leq c \cdot n$ , wobei  $c = c(m)$  konstant ist.

Die Annahme ist ok, falls  $T(n) \leq$

$$T\left(\left\lceil \frac{n}{m} \right\rceil\right) + T\left(\left\lfloor \frac{3}{4}n \right\rfloor\right) + \left\lceil \frac{n}{m} \right\rceil C_m + \left\lfloor \frac{n}{2} \right\rfloor \leq cn; \text{ dies gilt, falls}$$

$$\left\lceil \frac{n}{m} \right\rceil c + \left\lfloor \frac{3}{4}n \right\rfloor c + \left\lceil \frac{n}{m} \right\rceil C_m + \left\lfloor \frac{n}{2} \right\rfloor \leq cn \quad | \cdot \frac{1}{n} \quad (\text{IA})$$

$$\Leftrightarrow (\text{bis auf } \lceil \cdot \rceil, \lfloor \cdot \rfloor) \frac{c}{m} + \frac{3}{4}c + \frac{C_m}{m} + \frac{1}{2} \leq c$$

$$\Leftrightarrow -\frac{c}{m} - \frac{3}{4}c + c \geq \frac{C_m}{m} + \frac{1}{2}$$

$$\Leftrightarrow c \geq \frac{\frac{C_m}{m} + \frac{1}{2}}{1 - \frac{3}{4} - \frac{1}{m}}$$

**Bemerkung:**  $m = 11 \rightsquigarrow c = c(m) \approx 20$ . □

## Literatur:



Vaughan R. Pratt, Frances F. Yao:

*On lower bounds for computing the  $i$ -th largest element*

Proc. 14th Ann. IEEE SWAT, pp. 70–81 (1973)



Manuel Blum, Robert W. Floyd, Vaughan R. Pratt, Ron L. Rivest, Robert E. Tarjan:

*Time bounds for selection*

J. Comput. Syst. Sci. **7**, pp. 448–461 (1973)

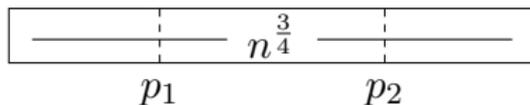
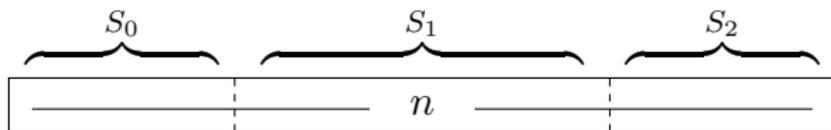
### 3. Randomisierter Median-Algorithmus

**Problemstellung:** Bestimme den Median von  $n$  Elementen

- 1 Wähle  $n^{\frac{3}{4}}$  Elemente zufällig und gleichverteilt aus den  $n$  Elementen aus.
- 2 Sortiere diese  $n^{\frac{3}{4}}$  Elemente mit einem (Standard-)  $n \log n$ -Algorithmus.
- 3 Setze

$p_1 := \max\{\frac{n^{\frac{3}{4}}}{2} - \sqrt{n}, 1\}$ -kleinstes Element der  $n^{\frac{3}{4}}$  Elemente.

$p_2 := \min\{\frac{n^{\frac{3}{4}}}{2} + \sqrt{n}, n^{\frac{3}{4}}\}$ -kleinstes Element der  $n^{\frac{3}{4}}$  Elemente.



- 4 Partitioniere die  $n$  Elemente in

$$S_0 := \{\text{Elemente} < p_1\}$$

$$S_1 := \{p_1 \leq \text{Elemente} \leq p_2\}$$

$$S_2 := \{p_2 < \text{Elemente}\}$$

- 5 Falls  $|S_0| \geq \lceil \frac{n}{2} \rceil$  oder  $|S_2| \geq \lceil \frac{n}{2} \rceil$  oder  $|S_1| \geq 4 \cdot n^{\frac{3}{4}}$ , dann wiederhole den Algorithmus;  
ansonsten sortiere  $S_1$  und liefere das  $(\lceil \frac{n}{2} \rceil - |S_0|)$ -kleinste Element davon ab.

## Satz 80

*Obiger randomisierter Algorithmus bestimmt den Median von  $n$ -Elementen mit einer erwarteten Anzahl von  $\frac{3}{2}n + o(n)$  Vergleichen.*

Beweis:

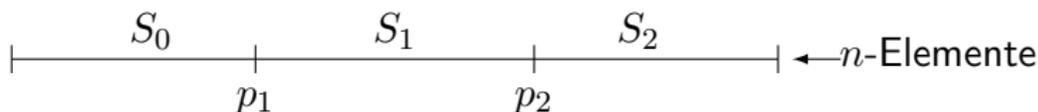
- i) Korrektheit: klar.

## Beweis (Forts.):

ii) Anzahl der Vergleiche in einer Iteration:

$$\mathcal{O}(n^{\frac{3}{4}} \log n^{\frac{3}{4}}) + \text{Kosten der Partitionierung}$$

Für die Partitionierung ist der naive Ansatz zu ungünstig, stattdessen:



Wähle zuerst jeweils mit Wahrscheinlichkeit  $\frac{1}{2}$  aus, ob Element  $x$  mit  $p_1$  oder  $p_2$  verglichen wird, mache **zweiten** Vergleich nur, falls nötig.

## Beweis (Forts.):

Die erwartete Anzahl von Vergleichen ist dann

$$\begin{aligned} &= \frac{n}{2} \left( \frac{|S_0|}{n} \cdot 1 + \frac{|S_1| + |S_2|}{n} \cdot 2 \right) + \frac{n}{2} \left( \frac{|S_2|}{n} \cdot 1 + \frac{|S_0| + |S_1|}{n} \cdot 2 \right) \\ &= \frac{n}{2} \left( \frac{|S_0| + |S_2|}{n} + 2 \overbrace{\frac{|S_0| + |S_1| + |S_2| + |S_1|}{n}}^n \right) \\ &= \frac{n}{2} \left( 3 + \frac{|S_1|}{n} \right) = \frac{3}{2}n + o(n) \end{aligned}$$

Wir zeigen nun, dass der Algorithmus mit Wahrscheinlichkeit  $\geq 1 - \mathcal{O}(n^{-\frac{1}{4}})$  nur eine Iteration benötigt (daraus folgt dann, dass insgesamt die Anzahl der Vergleiche  $\leq \frac{3}{2}n + o(n)$  ist).

## Beweis (Forts.):

Dafür verwenden wir Hilfsmittel aus der Wahrscheinlichkeitstheorie/Stochastik:

- **Bernoulli-Zufallsvariable (ZV):**  $X$ , Werte  $\in \{0, 1\}$  mit

$$X = \begin{cases} 1 & \text{mit WS } p \\ 0 & \text{mit WS } q = 1 - p \end{cases}$$

- **Erwartungswert** einer ZV:

$$\mathbb{E}[X] = \sum_{x \in \text{Wertebereich}} x \cdot \Pr[X = x]$$

( $X$  ist diskret, d.h. der Wertebereich von  $X$  ist endlich)

- **Markov-Ungleichung:**  $\Pr[X \geq t] \leq \frac{\mathbb{E}[X]}{t}$  für  $X$  nicht negativ
- **Chebyshev-Ungleichung:**  $\Pr[|X - \mathbb{E}[X]| \geq t] \leq \frac{\text{Var}(X)}{t^2}$

## Beweis (Forts.):

- **Binomialverteilung:** Seien  $X_1, \dots, X_n$  unabhängige, identisch verteilte Bernoulli-Zufallsvariablen mit  $\Pr[X_i = 1] = p$ .

$$X := \sum_{i=1}^n X_i .$$

$X$  ist binomial verteilt, mit Wertebereich  $\{0, 1, \dots, n\}$ .

$$\Pr[X = k] = \binom{n}{k} p^k (1 - p)^{n-k}$$

$$\mathbb{E}[X] = n \cdot p$$

$$\text{Var}[X] = n \cdot p \cdot (1 - p) = n \cdot p \cdot q .$$

In Zeichen:  $X \sim B(n, p)$

## Beweis (Forts.):

Die Auswahl der  $n^{\frac{3}{4}}$  Elemente wird wiederholt, falls  $|S_0| \geq \frac{n}{2}$ . Dies passiert gdw wir höchstens  $\frac{1}{2}n^{\frac{3}{4}} - \sqrt{n}$  Elemente aus der Hälfte aller Elemente  $\leq$  dem Median auswählen.

Wir bestimmen die Wahrscheinlichkeit dafür, dass keine neue Auswahl der  $n^{\frac{3}{4}}$  Elemente stattfinden muss.

Setze Bernoulli-Zufallsvariable  $X_1, \dots, X_n$  mit:

$$X_i = \begin{cases} 1, & \text{falls Element } i < \text{Median ausgewählt wird} \\ 0, & \text{sonst} \end{cases}$$

$X := \sum X_i$  ist binomialverteilt mit Parametern  $n$  und  $\frac{1}{2}n^{-\frac{1}{4}}$ , und  $\mathbb{E}[X] = \frac{1}{2}n^{\frac{3}{4}}$ ,  $\text{Var}[X] = n \cdot \frac{1}{2}n^{-\frac{1}{4}}(1 - \frac{1}{2}n^{-\frac{1}{4}}) = \frac{1}{2}n^{\frac{3}{4}}(1 - o(1))$ .

## Beweis (Forts.):

Die Wahrscheinlichkeit hierbei ist

$$\begin{aligned}\Pr[|S_0| \geq \frac{n}{2}] &= \Pr[X \leq \frac{1}{2}n^{\frac{3}{4}} - \sqrt{n}] \leq \Pr[|X - \mathbb{E}[X]| \geq \sqrt{n}] \\ &\leq \frac{\frac{1}{2}n^{\frac{3}{4}}(1 - o(1))}{n} \leq \frac{1}{2}n^{-\frac{1}{4}}(1 - o(1))\end{aligned}$$

Die anderen beiden Wahrscheinlichkeitsbedingungen ( $\Pr[|S_2| \geq \lceil \frac{n}{2} \rceil]$  und  $\Pr[|S_1| \geq 4 \cdot n^{\frac{3}{4}}]$ ) ergeben analoge Abschätzungen.

Damit: Wiederholung mit  $WS \leq \mathcal{O}(n^{-\frac{1}{4}})$ . □