5.7 Suchen

Wir betrachten eine Menge $X=\{x_1,\dots,x_n\}$ von n paarweise verschiedenen Zahlen.

Definition (Rang k Element)
Ein Element $x \in X$ ist von Rang k (heißt k-tes Element), wenn $|\{y \in X \mid y \leq x\}| = k$. Speziell heißt das $\lceil \frac{|X|}{2} \rceil$-Element "Median".

Es stellt sich folgende Frage:
Für gegebenes X und k, wie kann man das k-te Element schnell bestimmen?

Ein erster Ansatz ist schnell gefunden:

1. Sortiere X aufsteigend.
2. Gib das k-te Element zurück.
   (z.B. Mit linearem Durchlauf)

Die Korrektheit ist schnell zu sehen.
Die Laufzeit beträgt $\underbrace{O(n \log n)}_{\text{sortieren}} + \underbrace{O(k)}_{\text{Durchlauf zum } k\text{-ten Element}} = O(n \log n)$

Geht das schneller? Ja!
Mit dem Select-Algorithmus

Select(X, k)

1. if $(|X| \leq 5)$ then

1.1 Sortiere X

1.2 return k-tes Element (wie vorher)

2. else

2.1. Unterteile X in 5er Blöcke $B_1, \dots, B_t$ mit $t = \lceil \frac{n}{5} \rceil$

2.2. Bestimme den Median m_i in jedem Block B_i (z.B. durchsortieren)

2.3. Setze $M' := \bigcup_{i=1}^{t} \{m_i\}$

2.4. $m \leftarrow \text{Select}(M', \lceil \frac{t}{2} \rceil)$

2.5. $X_1 = \{x \in X \mid x < m\}$

2.6. $X_2 = \{m\}$

2.7. $X_3 = \{x \in X \mid x > m\}$

2.8. if $(|X_1| \geq k)$ then

2.8.1 return Select(X_1, k)

2.9 else if $(|X_1| + |X_2| \geq k)$ then

2.9.1 return m

2.10 else

2.10.1 return Select$(X_3, k - (|X_1| + |X_2|))$

Man erkennt: Select hat maximal 2 rekursive Aufrufe: 2.4 und entweder 2.8.1, 2.10.1 oder keinen weiteren.

Beispiel: $n = 25$, $k = 23$

$X = \{1, 22, 10, 13, 24, 6, 18, 21, 4, 25, 11, 16, 2, 20, 8, 17, 5, 12, 19, 14, 3, 9, 15, 7, 23\}$

1. Aufteilen in 5er Blöcke:

1	6	11	17	3
22	18	16	5	9
10	21	2	12	15
13	4	20	19	7
24	25	8	14	23

$\longrightarrow$

1	4	2	5	3
10	6	8	12	7
13	18	11	14	9
22	21	16	17	15
24	25	20	19	23

$\Rightarrow M' = \{13, 18, 11, 14, 9\}$

$\hookrightarrow$ Rekursiver Aufruf Select$(M', 3)$

Da $|M'| \leq 5 \Rightarrow$ erster Ansatz. Return 13

$\Rightarrow m = 13$

$\Rightarrow X_1 = \{1, 10, 4, 6, 2, 8, 11, 5, 12, 3, 7, 9\}$

$X_2 = \{13\}$

$X_3 = \{22, 24, 18, 21, 25, 16, 20, 14, 17, 19, 15, 23\}$

$\Rightarrow |X_1| = 12 < 23 \Rightarrow$ nicht 2.8.1

$|X_1| + |X_2| = 13 < 23 \Rightarrow$ nicht 2.9.1

$\Rightarrow$ Rekursiver Aufruf: Select$(X_3, 10)$

$\rightarrow X = X_3,\ k = 10$

$X = \{22, 24, 18, 21, 25, 16, 20, 14, 17, 19, 15, 23\}$

Aufteilen:

22	16	15
24	20	23
18	14	
21	17	
25	19	

$\xrightarrow{\text{sortieren}}$

18	14	15
21	16	23
22	17	
24	19	
25	20	

$\Rightarrow M' = \{22, 17, 15\}$

↳ Rekursiver Aufruf: Select$(M', 2)$

$\rightarrow |M'| \leq 5 \rightarrow$ return 17

$\rightarrow m = 17$

$\Rightarrow X_1 = \{14, 16, 15\}$

$X_2 = \{17\}$

$X_3 = \{18, 21, 22, 24, 25, 19, 20, 23\}$

$\Rightarrow$ Rekursiver Aufruf

Select$(X_3, 6)$

$\rightarrow X_3 = X_3,\ k = 6$

$X = \{18, 21, 22, 24, 25, 19, 20, 23\}$

Aufteilen:

18	19
21	20
22	23
24	
25	

Sortieren $\Rightarrow$

18	19
21	20
22	23
24	
25	

$\Rightarrow M' = \{22, 20\} \Rightarrow m = 20$

$\Rightarrow X_1 = \{18, 19\}$

$X_2 = \{20\}$

$X_3 = \{21, 22, 24, 25, 23\}$

$\Rightarrow$ Rekursiver Aufruf: Select$(X_3, 3)$

Da $|X_3| \leq 5 \Rightarrow$ erster Ansatz:

return 23

$\Rightarrow m = 23$

Das Konzept hier: Partitionierung von X in X_1, X_2, X_3 bereits bei Quicksort gesehen

$$\big|\; <m \;\big|\; \underbrace{m \;\big|\; >m}_{\geq m} \;\big|$$