Übungszettel 13, Aufgabe 1: Quickselect (Auswahl des k-kleinsten Element)

Gegeben: Feld der Länge n und ganze Zahl k mit $1 \leq k \leq n$.
Gesucht: k-kleinste Zahl im Feld

Ansatz 1: k Mal kleinstes Element ausgeben. $\mathcal{O}(k \cdot n)$
oder $n+1-k$ Mal größtes Element ausgeben. $\mathcal{O}((n-k)\cdot n)$ } Worst-Case $k \approx \frac{n}{2}$ $\Rightarrow \mathcal{O}(n^2)$ Laufzeit

Ansatz 2: Feld sortieren, dann k-tes Element ausgeben. $\mathcal{O}(n \cdot \log n)$ Laufzeit

Linearzeitansatz: Wie bei Quicksort wählen wir ein Pivotelement x und teilen das Feld in die drei Teilfelder „$<x$", „$=x$" und „$>x$" auf. Im Gegensatz zu Quicksort, können je zwei der drei Teilfelder verworfen werden.

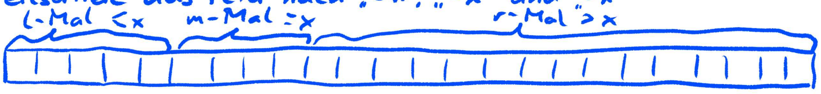
Wähle eine ganze Zahl i. In der Vorlesung wurde $i=2$ (also $2i+1=5$) gewählt.

Definiere $T(n)$ = Worst-Case-Zeit um aus n Elementen das k-kleinste mit dem folgenden Algorithmus auszugeben
($T(n)$ hängt implizit von i ab)

Teile das Feld in $\frac{n}{2i+1}$ Teilfelder der Länge $2i+1$. Bestimme die Mediane der Teilfelder.
(Da i konstant ist, dauert das jeweils $\mathcal{O}(1)$ und daher insgesamt $\mathcal{O}(n)$)

Bestimme den Median der Mediane. Das dauert maximal $T(\frac{n}{2i+1})$ Zeit.

Teilsortiere das Feld nach „$<x$", „$=x$" und „$>x$"



Wenn $k \leq l$ gilt, gebe das k-kleinste Element im Restfeld „$<x$" aus

Wenn $l < k \leq l+m$ gilt, gebe x aus.

Wenn $l+m < k$ gilt, gebe das $(k-(l+m))$-kleinste Element im Restfeld „$>x$" aus.

Größe des Restfeldes „$>x$":

Maximal alle $2i+1$ Elemente aus den $\frac{1}{2}\cdot\frac{n}{2i+1}$ Teilfeldern mit Median $>x$ $\Rightarrow$ Maximal $\frac{1}{2}n$ Elemente
+ Maximal die großen i Elemente aus den $\frac{1}{2}\cdot\frac{n}{2i+1}$* Teilfeldern mit Median $\leq x$ $\Rightarrow$ Maximal $\frac{i}{2(2i+1)}n$ Elemente

*: Wenn es Teilfelder mit Median $=x$ gibt, gibt es weniger Teilfelder mit Median $>x$, sodass in diesem Fall das Maximum kleiner wird. Alternativ können wir auch annehmen, dass alle Elemente verschieden sind.

Die Laufzeit $T(n)$ kann jetzt abgeschätzt werden:

$$T(n) \leq \underbrace{T\left(\frac{n}{2i+1}\right)}_{\text{Finden des Medians der Mediane}} + \underbrace{T\left(\left(\frac{1}{2}+\frac{i}{2(2i+1)}\right)n\right)}_{\text{Quickselect im Restfeld}} + \underbrace{c \cdot n}_{\text{Finden der Teilmediane und der Teilsortierung}}$$

(für geeignetes c)
(c hängt von i ab)

Zu zeigen: Es existiert ein c', sodass für alle n die Ungleichung $T(n) < c'n$ gilt.

Für festes n_0 gilt: Es gibt ein hinreichend großes $c' > 100c$, sodass für alle $1 \leq n < n_0$ die Ungleichung $T(n) < c'n$ gilt.
$\Rightarrow$ Induktionsanfang gilt

Induktionsschritt:

für $2i+1=7$ $(i=3)$:

$$T(n) \leq T\left(\tfrac{1}{7}n\right) + T\left(\left(\tfrac{1}{2}+\tfrac{3}{14}\right)n\right) + cn \overset{\text{Induktionsvoraussetzung}}{<} \tfrac{1}{7}c'n + \tfrac{5}{7}c'n + \tfrac{1}{100}c'n \leq \Big(\underbrace{\tfrac{1}{7}+\tfrac{5}{7}+\tfrac{1}{100}}_{<1}\Big)c'n < c'n$$

$\Rightarrow$ Für $i=3$ läuft der Algorithmus in Linearzeit.

für $2i+1=3$ $(i=1)$

$$T(n) \leq T\left(\tfrac{1}{3}n\right) + T\left(\left(\tfrac{1}{2}+\tfrac{1}{6}\right)n\right) + cn \overset{\text{I.V.}}{<} \tfrac{1}{3}c'n + \tfrac{2}{3}c'n + cn = c'n + \underbrace{cn}_{>0}$$

$\Rightarrow$ Für $i=1$ können wir die Linearzeit so nicht beweisen.

(Tatsächlich brauchen wir für $i=1$ $\Theta(n \log n)$ Zeit)