

Aufgabe 17: Quickselect

Man kann die Idee von QUICKSORT und dessen Prozedur SPLIT dazu benutzen, einen Algorithmus QUICKSELECT konstruieren, der für ein Feld A der Länge n und eine Zahl k mit $1 \leq k \leq n$ bei Aufruf von QUICKSELECT(A, k) das k -grösste Element von A zurückgibt – ohne A vollständig zu sortieren!

1. Formulieren Sie QUICKSELECT in pseudocode und zeigen Sie, dass die Anzahl der benötigten Vergleichsoperationen im *worst-case* quadratisch mit n wächst.
2. Bezeichnet $\bar{v}(n, k)$ die mittlere Anzahl von Vergleichen von QUICKSELECT(A, k) im Permutationsmodell und $\bar{v}(n) = \max_{1 \leq k \leq n} \bar{v}(n, k)$, so wird man (in Analogie zur Situation bei QUICKSORT) auf die Rekursions(un)gleichung

$$\bar{v}(n) \leq (n-1) + \frac{1}{n} \sum_{1 \leq k \leq n} \max\{\bar{v}(k-1), \bar{v}(n-k)\}$$

geführt, wobei verständlicherweise $\bar{v}(0) = \bar{v}(1) = 0$.

Erläutern Sie diese Ungleichung und zeigen Sie, dass sich daraus das asymptotische Verhalten $\bar{v}(n) \in O(n)$ ergibt.