
Grundlegende Algorithmen

Abgabe: bis 28. November, 16:00 Uhr, Briefkasten bei S0314

Hinweis: Es werden nur Abgaben von angemeldeten Studierenden korrigiert!

Aufgabe 1 (10 Punkte)

- (a) Wenden Sie QUICKSORT auf das Array $[2, 4, 1, 8, 6, 7, 9, 10, 3]$ an. Zeichnen Sie dabei das Array für jeden Partitionierungsschritt.
- (b) Finden Sie ein Array mit 7 Elementen das den Worst-Case von QUICKSORT darstellt (quadratisch viele Vergleiche) und begründen Sie, warum das von Ihnen gewählte Array den Worst-Case darstellt.
- (c) Wie muss partitioniert werden, damit QUICKSORT optimal läuft? Wenden Sie diese Strategie auf das Array aus (b) an.

Aufgabe 2 (10 Punkte)

Ein Element $A[i]$ eines Arrays A hat Rang $= k$ falls genau $k - 1$ Elemente aus A kleiner als $A[i]$ sind.

Eingabe: Array A mit n verschiedenen Elementen, $k \in \mathbf{N}$

Ausgabe: Index des Elements mit Rang k aus A

QUICKSELECT(A, k)

- ```
(1) $p = \text{PARTITION}(A, 1, n, n);$
(2) if $k = p$ then return p
(3) if $k < p$ then return QUICKSELECT($A[1..p - 1], k$)
(4) else
(5) return $p + \text{QUICKSELECT}(A[p + 1..n], k - p)$
(6)
```

Bestimmen Sie die Anzahl der Vergleiche, die QUICKSELECT bei einem Array der Länge  $n$  im Average Case benötigt. Nehmen Sie dabei an, dass alle Permutationen des Arrays mit gleicher Wahrscheinlichkeit auftreten. Die Wahrscheinlichkeit, dass das Pivotelement Rang  $k$  hat, ist daher  $\frac{1}{n}$ . Sie können wie folgt vorgehen:

1. Stellen Sie eine Rekursionsgleichung für  $V_{QS}(n, k)$  auf.
2. Eliminieren Sie  $k$  aus  $V_{QS}(n, k)$  durch Abschätzen ( $V(n) := \max\{\dots\}$ ).
3. Zeigen Sie mit Induktion, dass  $V(n) \leq 4n$  ist.