Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. Fabian Kuhn
P. Schneider, P. Bamberger, Y. Maus

Informatik 2 - Sommersemester 2018
Übungsblatt 2

Abgabe: Montag, 7. Mai, 14:00 Uhr

Aufgabe 1: $\mathcal{O}$-Notation *(6 Punkte)*

Beweisen oder widerlegen Sie die folgenden Aussagen. Nutzen Sie die Definitionen der $\mathcal{O}$-Notation.

(a) $100n \in \mathcal{O}(0.01n)$ *(1 Punkt)*

(b) $n \in \Omega(\log_2 3^n)$ *(1 Punkt)*

(c) $2n \in \mathcal{O}(10\sqrt{n})$ *(2 Punkte)*

(d) $f(n) + g(n) \in \Theta(\max\{f(n), g(n)\})$ für nicht negative Funktionen f und g. *(2 Punkte)*

Aufgabe 2: Sortieren nach Asymptotischem Wachstum *(6 Punkte)*

Sortieren Sie folgende Funktionen nach asymptotischem Wachstum. Schreiben Sie $g <_{\mathcal{O}} f$ falls $g \in \mathcal{O}(f)$ *und* $f \notin \mathcal{O}(g)$. Schreiben Sie $g =_{\mathcal{O}} f$ falls $f \in \mathcal{O}(g)$ *und* $g \in \mathcal{O}(f)$ (Kein Beweis nötig).

n^2	$\sqrt{n}$	2^n	$\log(n^2)$
3^n	n^{100}	$\log(\sqrt{n})$	$(\log n)^2$
$\log n$	$10^{100}n$	$n!$	$n \log n$
$n \cdot 2^n$	n^n	$\sqrt{\log n}$	n

Aufgabe 3: Quicksort Pivotwahl *(8 Punkte)*

Wir betrachten die folgende Variante des Quicksort-Algorithmus zum Sortieren eines Arrays A der Größe n. Der Algorithmus wählt in jedem Rekursionsschritt zunächst einen Index $p \in \{0, \dots, n-1\}$, welcher die Position des Pivotelements $x := A[p]$ angibt. Im selben Rekursionsschritt werden alle Elemente in A außer x auf zwei Teilarrays L und R aufgeteilt sodass L die Elemente kleiner gleich x, und R die Elemente echt größer x enthält.

Wir gehen davon aus, dass diese Aufteilung so durchgeführt wird, dass die ursprüngliche Ordnung zwischen den Elementen innerhalb der Teilarrays L bzw. R erhalten bleibt. Schließlich wird Quicksort rekursiv auf L und R ausgeführt und die sortierten Ergebnisarrays $L, \{x\}, R$ in einem Array zurückgegeben. Im Basisfall $n \leq 1$ wird kein Pivot gewählt sondern das Array zurückgegeben.

Gehen Sie vereinfachend davon aus, dass jeder Rekursionsschritt in genau n Zeitschritten durchgeführt wird wobei n die Größe des *aktuell* betrachteten Arrays ist. Sie dürfen zudem $n = 2^k - 1$ für ein $k \in \mathbb{N}$ annehmen. Wir betrachten zwei Strategien zur Wahl des Pivots: (1) $p = 0$ und (2) $p = \lfloor \frac{n}{2} \rfloor$.

(a) Beschreiben Sie für beide Strategien jeweils einen Input der Länge n, der den Best-Case bzw. den Worst-case bezüglich der Gesamtlaufzeit darstellt. (ohne Beweis) *(4 Punkte)*

(b) Stellen Sie für beide Strategien (1) und (2) für jeweils Worst-case und Best-case die Rekursiongleichungen $T_{1,w}(n), T_{1,b}(n), T_{2,w}(n), T_{2,b}(n)$ für die Laufzeit in Abhängigkeit von n auf. *(2 Punkte)*

(c) Schätzen Sie die Rekursionsgleichungen aus (b) möglichst scharf nach oben ab und geben Sie die asymptotische Laufzeitklasse an. *(2 Punkte)*