Präsenzblatt 4

Hinweis: Dieses Aufgabenblatt wurde von Tutor:innen erstellt. Die Aufgaben sind für die Klausur weder relevant noch irrelevant.

Aufgabe 4.1: Laufzeit von MergeSort

Führen Sie eine Laufzeitanalyse für MergeSort durch:

Algorithm 1: MergeSort

Function *MergeSort($A[a..b]$)*:
    if $b \leq a$ **then**
        return
    $m \leftarrow (a + b)$ div 2;
    MergeSort($A[a..m]$);
    MergeSort($A[m + 1..b]$);
    $B[a..b] \leftarrow A[a..b]$;
    Merge($B[a..m]$, $B[m + 1..b]$, $A[a..b]$)

Algorithm 2: Merge

Input: $A[a..m]$, $A[m + 1..b]$, $Z[1..b - a + 1]$
$i \leftarrow a$;
$j \leftarrow m + 1$;
for $k = 1$ **to** $b - a + 1$ **do**
    if $i > m$ **then**
        $Z[k] \leftarrow A[j]$;
        $j \leftarrow j + 1$
    else if $j > b$ **then**
        $Z[k] \leftarrow A[i]$;
        $i \leftarrow i + 1$
    else if $A[i] < A[j]$ **then**
        $Z[k] \leftarrow A[i]$;
        $i \leftarrow i + 1$
    else
        $Z[k] \leftarrow A[j]$;
        $j \leftarrow j + 1$

Tipp: Für Zweierpotenzen n kann man $T(n) \leq c(n \lceil \log_2 n \rceil + 1)$ induktiv beweisen ($T(n)$ bezeichne die rekurive Laufzeitfunktion und $c \in \mathbb{R}_+$ sei eine unabhängige Konstante). $\mathcal{O}(n \log_2 n)$ folgt dann, da $T(n)$ und $c(n \lceil \log_2 n \rceil + 1)$ monoton wachsen.

Aufgabe 4.2: Fakultät in die Schranken weisen

Beweisen Sie $\left(\frac{n}{2}\right)^{\frac{n}{2}} \leq n! \leq n^n$.

Aufgabe 4.3: Stabil und gut gebaut

Zeigen Sie, dass jeder instabile vergleichsbasierte Algorithmus U mit Laufzeit T_U in einen stabilen Algorithmus mit gleicher Laufzeit transformiert werden kann.

Aufgabe 4.4: Zweidrittel sortiert ist halb gewonnen

Gegeben sei folgender Sortieralgorithmus für Arrays A der Länge n:

1. Sortiere die ersten $\left\lceil\frac{2}{3}\right\rceil$ von A.
2. Sortiere die zweiten $\left\lceil\frac{2}{3}\right\rceil$ von A.
3. Sortiere die ersten $\left\lceil\frac{2}{3}\right\rceil$ von A.

Die innere Sortierung geschieht rekursiv.

Beweisen Sie, dass der Algorithmus korrekt ist. Stellen Sie das formale Kriterium der Korrektheit von Sortieralgorithmen auf.

Welche Laufzeit weist der Algorithmus auf?

Aufgabe 4.5: Binär zählen leicht gemacht

Analysieren Sie die Laufzeit, eine Binärzahl mit n Stellen 2^n mal zu inkrementieren. Welche amortisierte Laufzeit lässt sich für jede einzelne Operation feststellen?