Universität Bremen

FB3: Mathematik/Informatik

Prof. Dr. Daniel Neuen
Jens Schlöter

Wintersemester 2023/2024

Algorithmentheorie

Präsenzübung 2

Präsenzübung 2.1

Zeigt oder widerlegt folgende Aussagen.

(a) $n^2 \in o(5n^3 + 2n^2 + n + 3)$.

(b) $(\log_2(n))^3 \in \mathcal{O}(n)$.

(c) $3^n \in \mathcal{O}(2^n)$.

(d) Für alle $f, g : \mathbb{N} \to \mathbb{R}$ gilt: $f \in \mathcal{O}(g)$ oder $g \in \mathcal{O}(f)$.

Präsenzübung 2.2

Gebt für folgende Rekursionsgleichungen eine geschlossene Form in Θ-Notation an und zeigt die Korrektheit per Induktion. Für beide Rekursionsgleichungen könnt ihr annehmen, dass $T(1) = 1$.

(a) $T(n) = 2T(\frac{n}{4}) + 1$.

(b) $T(n) = 3T(\frac{n}{5}) + n \log(n)$.

Präsenzübung 2.3

Gegeben sei der Algorithmus auf der nächsten Seite, welcher als Eingabe ein Array $A = [a_0, \ldots, a_{n-1}]$ mit ganzen Zahlen bekommt. Was berechnet dieser Algorithmus? Was ist die Worst-Case Laufzeit?

Algorithmus 1 : Algorithmus für Präsenzübung 2.3

Input : Array $A = [a_0, \dots, a_{n-1}]$ mit $a_j \in \mathbb{Z}$ für alle $j \in \{1, \dots, n\}$.
Output : ?

$\ell = 0$
$r = n - 1$
return `divideAndConquer`(ℓ,r,A)

Function *`divideAndConquer`*(ℓ,r,A)
    if $\ell > r$ **then**
        return 0
    $m := \lfloor \frac{\ell + r}{2} \rfloor$
    $s_1 :=$ `divideAndConquer`(ℓ, m, A)
    $s_2 :=$ `divideAndConquer`$(m + 1, r, A)$
    $p_\ell := 0$
    $sum := 0$
    $i := m$
    while $i \geq \ell$ **do**
        $sum := sum + A[i]$
        if $sum > p_\ell$ **then** $p_\ell = sum$
        $i = i - 1$
    $p_r := 0$
    $sum := 0$
    $i := m + 1$
    while $i \leq r$ **do**
        $sum := sum + A[i]$
        if $sum > p_r$ **then** $p_r = sum$
        $i = i + 1$
    $s_3 := p_\ell + p_r$
    return $\max\{s_1, s_2, s_3\}$