



## Algorithmen und Datenstrukturen Sommersemester 2020 Übungsblatt 2

Abgabe: Mittwoch, 27.05.2020, 16:00 Uhr.

### Aufgabe 1: $\mathcal{O}$ -Notation

(7 Punkte)

Beweisen oder widerlegen Sie die folgenden Aussagen anhand der *Mengendefinition* der  $\mathcal{O}$ -Notation (Vorlesungsfolien Woche 2, Folie 11 und 12).

- (a)  $3n^3 + 8n^2 + n \in \mathcal{O}(n^3)$  (1 Punkt)
- (b)  $2^n \in o(n!)$  (2 Punkte)
- (c)  $2 \log n \in \Omega((\log n)^2)$  (2 Punkte)
- (d)  $\max\{f(n), g(n)\} \in \Theta(f(n) + g(n))$  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).

$\sqrt{n}$	$2^n$	$n!$	$\log(n^3)$
$3^n$	$n^{100}$	$\log(\sqrt{n})$	$(\log n)^2$
$\log n$	$10^{100}n$	$(n+1)!$	$n \log n$
$2^{(n^2)}$	$n^n$	$\sqrt{\log n}$	$(2^n)^2$

### Aufgabe 3: Stabiles Sortieren

(7 Punkte)

Ein Sortieralgorithmus heißt stabil, falls Elemente mit gleichem Schlüssel in der gleichen Reihenfolge bleiben. Man nehme z.B. an, man möchte folgende Strings sortieren wobei der Sortierschlüssel in diesem Beispiel der *Anfangsbuchstabe in alphabetischer Reihenfolge* ist:

[“tuv”, “adr”, “bbc”, “tag”, “taa”, “abc”, “sru”, “bcb”]

Ein *stabiler* Sortieralgorithmus muss dann folgenden Output generieren:

[“adr”, “abc”, “bbc”, “bcb”, “sru”, “tuv”, “tag”, “taa”]

Ein Sortieralgorithmus wäre nicht stabil (bzgl. diesem Sortierschlüssel), wenn er z.B. Folgendes ausgibt:

[“abc”, “adr”, “bbc”, “bcb”, “sru”, “taa”, “tag”, “tuv”]

- (a) Welche der in der Vorlesung gezeigten Sortieralgorithmen (ausser `CountingSort`) sind in der gegebenen Form *nicht* stabil? Beweisen Sie jeweils anhand eines geeigneten Beispiels. (4 Punkte)
- (b) Beschreiben Sie eine Methode, welche jeden Sortieralgorithmus stabil macht ohne die *asymptotische* Laufzeit zu ändern. Begründen Sie. (3 Punkte)