

Aufgabe 1 O-Notation

10 Punkte

- (a) Sortieren Sie die folgenden Laufzeiten aufsteigend. (Wenn $f(n) \in O(g(n))$, aber nicht $g(n) \in O(f(n))$, dann soll $f(n)$ vor $g(n)$ stehen.) Erklären Sie Ihre Antwort.

(1) n^3 (2) $\log_2 n$ (3) $1,8^n$ (4) n (5) 3^n (6) \sqrt{n} (7) $n(\log_2 n)^2$ (8) n^2

- (b) Angenommen, ein Programm hat Laufzeit $f(n)$, wobei $f(n)$ eine von Ihnen gewählte Funktion aus Teil (a) ist. Nehmen Sie weiter an, dass in zehn Jahren die Rechner 1000 mal schneller sind als heute. Probleme *welcher Größe* kann man dann in derselben Zeit lösen, die man heute für Probleme der Größe $n = 20$ braucht?

Aufgabe 2 Simultanes Minimum und Maximum

10 Punkte

- (a) Sei A eine Folge von n paarweise vergleichbaren Elementen. Geben Sie einen einfachen, möglichst effizienten Algorithmus zum Finden des Minimums und des Maximums von A . Bestimmen Sie die erforderliche Anzahl von Vergleichen im schlimmsten Fall als Funktion von n exakt.
- (b) Lösen Sie das gleiche Problem mit dem Teile-und-Herrsche Prinzip, indem Sie A in zwei Hälften zerlegen, falls $n \geq 2$ ist. Bestimmen Sie die benötigte Anzahl der Vergleiche im schlimmsten Fall exakt. Sie dürfen dabei annehmen, dass n eine Zweierpotenz ist.

Aufgabe 3 Einfache Datenstrukturen in Java

10 Punkte

- (a) Betrachten Sie den folgenden Entwurf für ein Interface für einen Stapel.

```
public interface Stack<E> {
    void push(E element);
    E pop();
    E top();
    int size();
    boolean isEmpty();
}
```

Vervollständigen Sie das Interface mit sinnvollen Ausnahmen. Schreiben Sie eine Klasse `ListStack<E>` und eine Klasse `DynamicArrayStack<E>`. Die zwei Klassen sollen `Stack` implementieren, jeweils mit einer verketteten Liste und mit einem dynamisch wachsenden Feld. Achten Sie auf eine geeignete Fehlerbehandlung und Ausnahmen und testen Sie Ihre Implementierung mit verschiedenen Eingaben.

- (b) Schreiben Sie ein Programm, welches für beliebige Zeichenketten, die verschiedene Klammern enthalten, überprüft, ob sie korrekt geklammert sind. Erlaubte Klammertypen sind `()`, `[]`, `<>` und `{}`. Verwenden Sie eine geeignete Datenstruktur. Zum Beispiel ist `"(A){L[P]3}"` korrekt geklammert, nicht aber `"(A){L[P]3}"` und `"(A){L[P3}"`.

Hinweis: Einzelne Zeichen eines Strings bekommt man in Java mit `charAt`.

- (c) (*freiwillig, 5 Zusatzpunkte*) Informieren Sie sich über das Konzept des *Modultests* (unit testing) und insbesondere über das *JUnit*-Framework. Laden Sie das JUnit-Framework herunter und implementieren Sie jeweils 5 Testcases für Ihre beiden Stapelimplementierungen aus (a) und 2 Testcases für Ihr Programm aus (b).