
B1 Entwurf / Analyse von Programmen und Programmiertechniken

B1.1 Schätzungen zufolge werden 25-30% der weltweiten Rechenzeit für das Sortieren aufgewendet. Eines der einfachsten Sortierverfahren ist Bubblesort.

(a) Implementieren Sie den Algorithmus Bubblesort in der Methode

```
bubbleSort(daten: int[]).
```

Der Sortieralgorithmus Shakersort (auch als „Cocktailsort“ bekannt) ist eine Variante von Bubblesort. Während Bubblesort das Array immer von links nach rechts bearbeitet, wechselt Shakersort nach jedem Durchgang die Richtung. Für den Rest dieser Aufgabe wird angenommen, dass sowohl Bubblesort als auch Shakersort abbrechen, sobald ein Durchgang ohne Vertauschungen abgeschlossen wurde.

- (b) Stellen Sie den Inhalt des Arrays { 45, 12, 99, 37, 2 } nach jeder Vertauschungsoperation des Shakersort-Algorithmus dar. Dabei erfolgt der erste Durchlauf von links nach rechts.
- (c) Erläutern Sie den Begriff „worst case“ im Zusammenhang mit Sortieralgorithmen.
- (d) Begründen Sie, warum das Array { 2, 3, 4, 5, ..., 99, 100, 1 } für Bubblesort einen worst case in Bezug auf die Anzahl der Vergleiche darstellt, für Shakersort aber nicht.

(8 VP)

B1.2 Ein effizienteres Verfahren zum Sortieren von unsortierten Daten ist Mergesort.

(a) Beschreiben Sie, wie der Algorithmus Mergesort arbeitet.

| IntListenknoten |
|--|
| ⊕ daten: Integer |
| ⊕ nachfolger: IntListenknoten |
| © IntListenknoten(daten: Integer, nachfolger: IntListenknoten) |

Mergesort kann auch auf verketteten Listen effizient ausgeführt werden. Dabei wird eine Hilfsmethode

```
merge(a: IntListenknoten, b: IntListenknoten): IntListenknoten
```

verwendet, die zwei sortierte Listen übergeben bekommt und diese zu einer einzigen sortierten Liste zusammenfügt und diese zurückgibt. Die Listen werden jeweils durch den Verweis auf ihren ersten Knoten repräsentiert.

(b) Erläutern Sie anhand des gegebenen Nassi-Shneiderman-Diagramms, wie die `merge`-Methode arbeitet.

Die Liste a habe die Länge n , die Liste b habe die Länge m . Es wird ein Aufruf $\text{merge}(a, b)$ ausgeführt.

$\text{merge}(a: \text{IntListenknoten}, b: \text{IntListenknoten}): \text{IntListenknoten}$

| | |
|--|--|
| ja a = null ? nein | |
| gib b zurück | \emptyset |
| ja b = null ? nein | |
| gib a zurück | \emptyset |
| a.daten <= b.daten ? | |
| ja | nein |
| a.nachfolger = merge(a.nachfolger, b) | b.nachfolger = merge(a, b.nachfolger) |
| gib a zurück | gib b zurück |

- (c) Analysieren Sie die Methode `merge` in Bezug auf die Anzahl der rekursiven Aufrufe in Abhängigkeit von n und m . Geben Sie für die Anzahl eine Unter- und eine Obergrenze an.

(7 VP)

Meta 1: Erläuterungen zu Nassi-Shneiderman-Diagrammen oder Quelltexten (Pseudocode) können gefragt werden. Es ist nicht gedacht, zeilenweise zu übersetzen, was da steht, sondern die Wirkungsweise im Zusammenhang und den Aufbau des Codes zu erklären.

Meta 2: Es wird nicht verlangt, selbst Nassi-Shneiderman-Diagramme oder Pseudocode zu erstellen.

Meta 3: Dies ist ein Beispiel für das Mischen von Themengebieten.

Meta 4: Es kann Aufgaben geben, in denen keine OOP vorkommt. Es kann aber auch sein, dass wie in den früheren Jahren reine OOP-Aufgaben gestellt werden.