

Exercise Sheet 5 - Gruppe 4

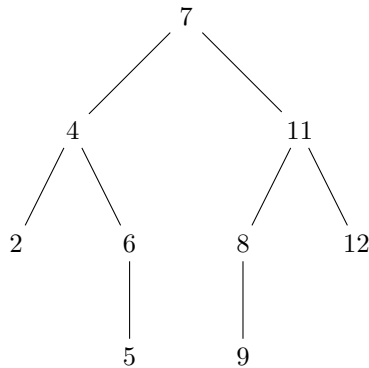
Jeudyl Robles Pidiache

November 26, 2023

4/10 Punkte

Exercise 1: 3/3p.

a)



- Postorder : 2, 5, 6, 4, 9, 8, 12, 11, 7 ✓
- Preorder : 7, 4, 2, 6, 5, 11, 8, 9, 12 ✓
- Inorder : 2, 4, 5, 6, 7, 8, 9, 11, 12 ✓

Exercise 2:

a) 0/3p.

Ursprüngliches Array

$A = [12, 10, 3, 21, 17, 5, 9]$

MergeSort-Prozess

Analog zu Vorlesungsslides!

1. **Erste Ebene der Teilung** - A in zwei Hälften teilen: $[12, 10, 3]$ und $[21, 17, 5, 9]$.

2. Zweite Ebene der Teilung

- Die erste Hälfte in $[12, 10, 3]$, in $[12, 10]$ und $[3]$ teilen.
- Die zweite Hälfte in $[21, 17, 5, 9]$, in $[21, 17]$ und $[5, 9]$ teilen.

3. Dritte Ebene der Teilung

- Für $[12, 10]$, in $[12]$ und $[10]$ teilen.
- Für $[21, 17]$, in $[21]$ und $[17]$ teilen.
- Die Arrays $[3]$, $[5, 9]$ sind bereits in ihrer kleinsten Form.

4. Beginn des Zusammenführens und Sortierens

- $-[12]$ und $[10]$ zu $[10, 12]$ zusammen führen.
- $-[21]$ und $[17]$ zu $[17, 21]$ zusammen führen.
- $-[10, 12]$ und $[3]$ zu $[3, 10, 12]$ zusammen führen.
- $-[5]$ und $[9]$ zu $[5, 9]$ zusammen führen.
- $-[17, 21]$ und $[5, 9]$ zu $[5, 9, 17, 21]$ zusammen führen.
- $-[3, 10, 12]$ und $[5, 9, 17, 21]$ zu $[3, 5, 9, 10, 12, 17, 21]$ zusammen führen.

Ergebnis

Sortiertes A = $[3, 5, 9, 10, 12, 17, 21]$

| Step | Array |
|---|---------------------------------------|
| Ursprüngliches Array | $[12, 10, 3, 21, 17, 5, 9]$ |
| Erste Ebene der Teilung | $[12, 10, 3], [21, 17, 5, 9]$ |
| Zweite Ebene der Teilung | $[12, 10], [3], [21, 17], [5, 9]$ |
| Dritte Ebene der Teilung | $[12], [10], [3], [21], [17], [5, 9]$ |
| Beginn des Zusammenführens und Sortierens | $[10, 12], [3], [17, 21], [5, 9]$ |
| | $[3, 10, 12], [5, 9, 17, 21]$ |
| Ergebnis | $[3, 5, 9, 10, 12, 17, 21]$ |

Table 1: MergeSort-Prozess

NICHT
parallel
auf
linker und
rechter Seite
arbeiten.

b)

Die Liste

$L = [10, 7, 5, 15, 13, 3, 6, 11, 4]$

falsche Darstellung und falsche Durchführung

QuickSort-Prozess

1. **Step:** [4, 5] - Sortiert die Teilliste links vom Pivotwert 6.
2. **Step:** [10, 11] - Sortiert die Teilliste rechts vom Pivotwert 7.
3. **Step:** [7, 10, 11] - Kombiniert die sortierte Liste mit dem Pivotwert 7.
4. **Step:** [4, 5, 6, 7, 10, 11] - Kombiniert die sortierte Liste mit dem Pivotwert 6.
5. **Step:** [3, 4, 5, 6, 7, 10, 11] - Kombiniert die sortierte Liste mit dem Pivotwert 3.
6. **Step:** [3, 4, 5, 6, 7, 10, 11, 13, 15] - Kombiniert die sortierte Liste mit dem Pivotwert 13.

Ergebnis

Endgültige Liste: [3, 4, 5, 6, 7, 10, 11, 13, 15].

| Step | Array |
|---------------------------------------|---|
| Die Liste | [10, 7, 5, 15, 13, 3, 6, 11, 4] |
| Step: [4, 5] | Sortiert die Teilliste links vom Pivotwert 6 |
| Step: [10, 11] | Sortiert die Teilliste rechts vom Pivotwert 7 |
| Step: [7, 10, 11] | Kombiniert die sortierte Liste mit dem Pivotwert 7 |
| Step: [4, 5, 6, 7, 10, 11] | Kombiniert die sortierte Liste mit dem Pivotwert 6 |
| Step: [3, 4, 5, 6, 7, 10, 11] | Kombiniert die sortierte Liste mit dem Pivotwert 3 |
| Step: [3, 4, 5, 6, 7, 10, 11, 13, 15] | Kombiniert die sortierte Liste mit dem Pivotwert 13 |
| Ergebnis | [3, 4, 5, 6, 7, 10, 11, 13, 15] |

Table 2: QuickSort-Prozess

c) *1p.*

Instabilität von Sortieralgorithmen

Ein Sortieralgorithmus ist instabil, wenn er die relative Reihenfolge gleichwertiger Elemente verändert. QuickSort ist ein Beispiel für einen instabilen Sortieralgorithmus. Um dies zu demonstrieren, kann man ein kleines Array verwenden, in dem die Elemente nicht nur einen Wert, sondern auch einen zusätzlichen Identifikator haben, um ihre ursprüngliche Reihenfolge zu kennzeichnen.

Beispiel

Nehmen wir das Array:

$(2_A, 3, 2_B, 1)$ ✓

Hierbei sind "2_A" und "2_B" gleichwertige Elemente (beide haben den Wert 2), aber sie haben unterschiedliche Identifikatoren (A und B), um sie unterscheiden zu können. *sieht*

Nach dem Sortieren mit einem instabilen Algorithmus wie QuickSort ~~könnte~~ das Array ~~so aussehen:~~ *so aus:*

$(1, 2_B, 2_A, 3)$ ✓

In diesem sortierten Array sehen wir, dass die Reihenfolge von "2_A" und "2_B" vertauscht wurde, obwohl sie gleichwertig sind. Dies zeigt, dass QuickSort ein instabiler Sortieralgorithmus ist.