

Übungsblatt 06

Aufgabe T20

Beweisen Sie, dass es keinen vergleichsbasierten Sortieralgorithmus gibt, welcher ein beliebiges Array der Größe fünf mit nur sechs Vergleichen sortieren kann.

Aufgabe T21

Gegeben ist folgende Zahlenfolge: 8, 6, 3, 7, 4, 2, 20, -45

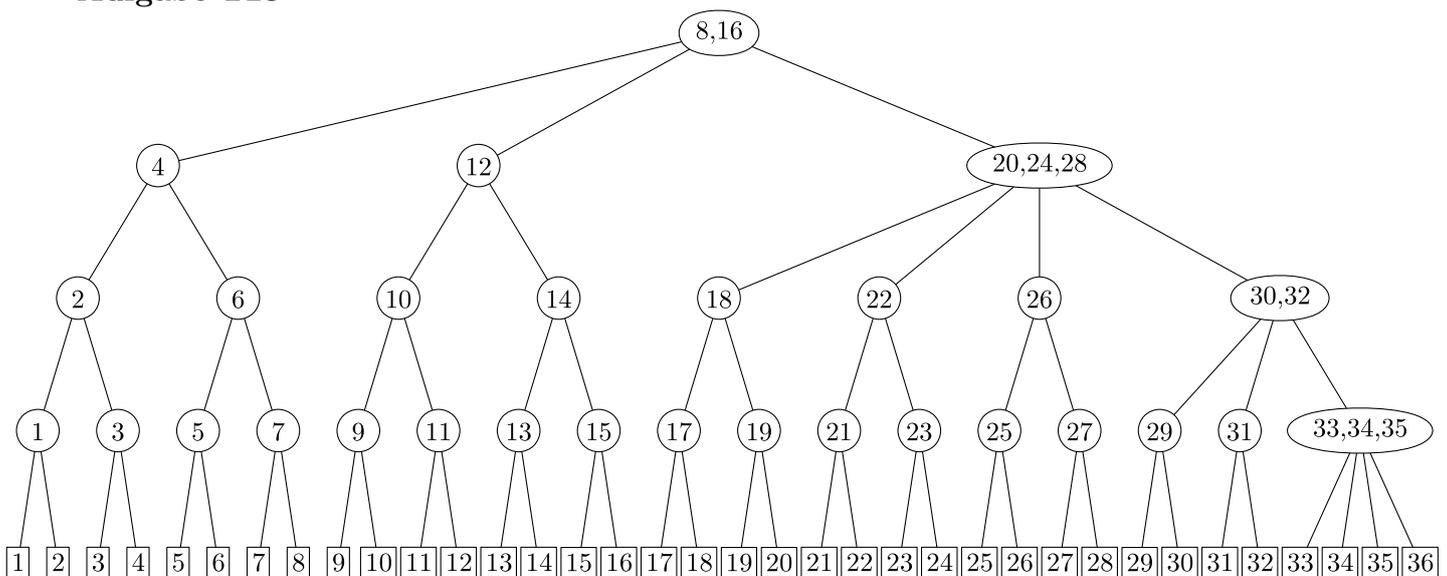
- Wie viele Inversionen hat diese?
- Wie viele Läufe hat diese? Ein Lauf ist eine maximale sortierte Teilfolge.
- Was macht Quicksort in der ersten Partitionierungsphase daraus?
- Was macht Mergesort in der letzten Mischphase?

Aufgabe T22

Erfinden Sie ein Sortierverfahren, das ein Array der Größe n in linearer Zeit sortieren kann unter der Annahme, dass das Array ausschließlich ganze Zahlen zwischen -1000 und 1000 enthält. Ist ihr Verfahren In-place?

Erläutern Sie die Idee und stellen Sie den Algorithmus zusätzlich in Form von Pseudocode oder einer Implementierung in einer vernünftigen Programmiersprache vor.

Aufgabe T23



Was passiert, wenn wir den Schlüssel 14 aus diesem (2, 4)-Baum löschen?

Aufgabe H16 (10 Punkte)

Hier ist noch einmal eine einfache Variante des Quicksort-Algorithmus:

```
procedure quicksort(L, R) :
  if R ≤ L then return fi;
  p := a[L]; l := L; r := R + 1;
  do
    do l := l + 1 while a[l] < p;
    do r := r - 1 while p < a[r];
    vertausche a[l] und a[r];
  while l < r;
  temp := a[r]; a[L] := a[l]; a[l] := temp; a[r] := p;
  quicksort(L, r - 1); quicksort(r + 1, R)
```

Das Array a enthalte die Zahlen 4, 1, 3, 2, 5, 9. Welche rekursiven Aufrufe gibt es? Geben Sie den Inhalt des Arrays jeweils am Anfang jedes Aufrufs und bevor die letzte Zeile ausgeführt wurde an.

Aufgabe H17 (10 Punkte)

Gegeben sei ein Array der Länge n , welches garantiert nur k verschiedene Zahlen enthält – jede aber beliebig oft. Wir gehen davon aus, dass n viel größer als k ist und k sogar viel kleiner als $\log n$ sein kann.

Erfinden und beschreiben Sie ein Sortierverfahren, welches in dieser speziellen Situation sehr schnell ist.

Genauer gesagt: Die Laufzeit soll nur $O(n \log k)$ betragen.

Aufgabe H18 (5+5 Punkte)

Gegeben sei das Array [3, 9, 6, 4, 8, 10, 5, 2].

- a) Führen Sie den Heapsort-Algorithmus aus der Vorlesung darauf aus.
- b) Führen Sie den Mergesort-Algorithmus aus der Vorlesung darauf aus.

Geben Sie alle Zwischenschritte an.