

Aufgabe 3 (10 Punkte)

Fügen Sie in einen anfänglich leeren AVL-Baum der Reihe nach die Elemente

30, 16, 37, 33, 38, 42, 35, 21, 16, 18, 31 und 34

ein und löschen Sie anschließend das Element 30. Geben Sie die entstehenden Bäume nach jedem Schritt, der Ausgleichsrotationen erfordert, sowie am Ende an. Geben Sie die jeweils durchgeführten Rotationen an.

Aufgabe 4 (10 Punkte)

Gegeben ist das folgende Programm, mit dem das Array Feld vorbelegt wird.

```
PROGRAM SortTest;

Const n = 128;    {muß eine Zweierpotenz sein}
VAR   Feld: ARRAY[0..n-1] OF INTEGER;
      i: INTEGER;

BEGIN
  i := 1;
  WHILE i < n DO
    BEGIN
      Feld[i-1] := i;
      FELd[i] := i-1;
      i := i+2
    END;

    {Sortierprogramm}

  END.
```

Anstelle des Kommentars Sortierprogramm sollen die unten aufgeführten Sortieralgorithmen auf die zuvor im Array erzeugten Daten angesetzt werden. Geben Sie jeweils die obere Schranke der asymptotischen Komplexität in O -Notation genau an und begründen Sie das Ergebnis kurz. Sie können vereinfachend annehmen, daß die Feldlänge eine Zweierpotenz ist.

- (a) SelectionSort
- (b) InsertionSort
- (c) BubbleSort optimiert (Das Verfahren bricht ab, wenn sich nichts mehr ändert.)
- (d) MergeSort
- (e) Quicksort