Aufgabe 1:

Gegeben ist der folgende Ausschnitt aus einem Rot-Schwarz Baum:

```
public class RSB1<K extends Comparable<K>,D> {

class Node {
      public Node(K key,D data) {..}

      K m_Key;
      D m_Data;
      Node m_Left = null;
      Node m_Right = null;
      boolean m_bIsRed = true;
} ...
```

Implementieren Sie eine Objektmethode boolean checkDepth(), die true zurückliefert, genau dann wenn die maximale Tiefe im Rot-Schwarz Baum nicht größer als zweimal der minimalen Tiefe im Baum ist.

Aufgabe 2:

Gegeben ist die folgende Implementierung einer Adjazenzmatrix für Graphen:

```
public class GraphMatrix {
      public GraphMatrix(int iNrOfNodes) {
            m_Matrix = new boolean[iNrOfNodes][iNrOfNodes];
      }
      private boolean[][] m_Matrix;
}
```

Implementieren Sie eine Methode int shortestPath(int start,int end),die die Länge des kürzesten Wegs von start zu end berechnet (Tipp: rekursiv lösen).

Aufgabe 3:

Gegeben sei das Array [5,9,8,7,10]. Zeichnen Sie die fünf Array Zustände, die während des Heapsorts auftreten (siehe Code unten).

```
static <K extends Comparable<K>> void heapsort(K[] field) {
      for(int i = (field.length-1) / 2; i >= 0; --i)
            downheap(field, field.length, i);
      print(field);
      for(int i = field.length-1; i > 0; --i) {
            K tmp = field[i];
            field[i] = field[0];
            field[0] = tmp;
            downheap(field, i, 0);
            print(field);
      }
}
```