Universität Konstanz
Fachbereich Informatik & Informationswissenschaft
Dr. Sven Kosub

Einführung in die Informatik 2
SS 2010

6. Übungsblatt

Ausgabe: 18.05.2010 **Abgabe:** 25.05.2010, vor der Vorlesung

Aufgabe 1: Binärbäume — 10 Punkte

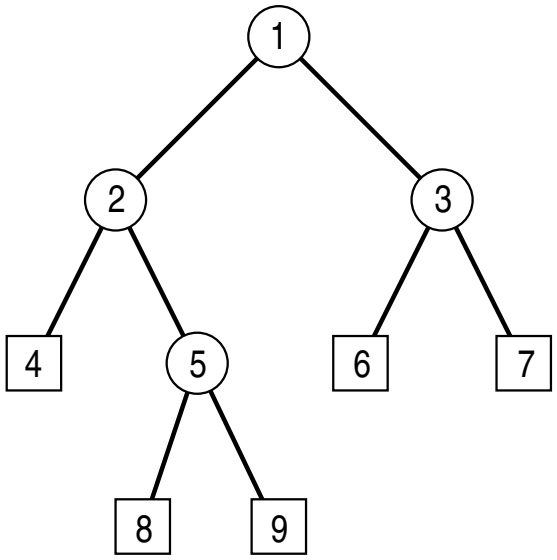
Ein voller Binärbaum T mit n Knoten kann wie folgt in einem Array abgespeichert werden:

- Die Wurzel von T steht an Position 1 im Array.
- Ist v ein Knoten von T an Position $P(v)$ im Array, so ist die Position des linken Kindes von v gerade $2 \cdot P(v)$ und die Position des rechten Kindes von v gerade $2 \cdot P(v) + 1$.

Die Position 0 im Array wird nicht benutzt.

(a) Wie groß müssen Sie das Array wählen, damit ein beliebiger voller Binärbaum mit n Knoten in Ihrem Array Platz findet?

(b) Geben Sie den Inhalt des Arrays an, das folgenden vollen Binärbaum abspeichert:



Aufgabe 2: Suchbäume — 10 Punkte

Betrachten Sie die in der Vorlesung behandelte Methode `insertItem` zum Einfügen neuer Objekte in einen binären Suchbaum.

(a) Welcher Suchbaum entsteht, wenn Sie die `int`-Objekte $3, 7, 13, 2, 11, 7, 1, 5, 17, 11, 9, 4$ in genau dieser Reihenfolge in einen anfangs leeren binären Suchbaum einfügen?

(b) Welcher Suchbaum entsteht, wenn Sie die Einfügereihenfolge aus Teilaufgabe (a) umkehren?

Aufgabe 3: Suchbäume **10 Punkte**

Binäre Suchäume wurden in der Vorlesung durch die Klasse `SearchTree` implementiert. Diese Klasse benutzt Objekte der wie folgt definierten Klasse `Node` als Knoten der Bäume:

```
public class Node {
   Comparable key;
   Node parent;
   Node leftChild;
   Node rightChild;
   boolean isExternal;
}
```

Entwerfen und implementieren Sie eine Methode

```
public Node inorderNext(Node n);
```

in der Klasse `SearchTree`, die zu einem inneren Knoten `n` den nächsten inneren Knoten in Inorder-Reihenfolge zurückgibt, falls solch ein Knoten existiert, und sonst `n`.