Prof. Dr. V. Linnemann
Nils Hoeller
Jana Neumann
Universität zu Lübeck
Institut für Informationssysteme

Lübeck, den 23. April 2009

Algorithmen und Datenstrukturen

Sommersemester 2009

3. Übungsblatt

Aufgabe 1: Rekursion und Backtracking

Backtracking ist eine Vorgehensweise, die vergleichbar ist mit dem Lernen durch Versuch und Irrtum. Ein typisches Beispiel für Probleme, die durch Backtracking gelöst werden können, ist das Finden eines Ausgangs in einem Labyrinth. Dabei ist eine gängige Vorgehensweise, sich den aktuell gegangen Weg mit einem Stück Kreide zu markieren. Endet man in einer Sackgasse, so kann man stets aufgrund der Markierung zum Ursprungsort zurückkommen und einen alternativen Weg ausprobieren. Irgendwann findet man so den Ausgang, falls solch einer existiert.
In dieser Aufgabe sollen Sie einen Backtracking-Algorithmus, der das Labyrinth-Problem löst, entwerfen und in Java implementieren.

- Laden Sie sich hierzu von der Übungswebseite die Programmdateien Labyrinth.java und Main.java herunter.
- In der Main.java finden Sie das eigentliche Programm, welches zunächst ein Beispiel-Labyrinth erzeugt und zeichnet. Hierbei wird das Labyrinth als Matrix (2D Array) mit folgender Codierung ausgegeben:
  - 'W' Wand '*' Kreidemarkierung ' ' Freier Weg 'A' Ausgang
- Das Beispiel-Labyrinth sieht dann folgendermaßen aus:

```
WWWWW
W W A
    W
WW WW
WW
```

- Die Methode suche_ausgang(int x,int y) in Labyrinth.java soll eine Aussage darüber treffen, ob ein Weg vom Startpunkt x,y zum Ausgang existiert. Sie wird im Hauptprogramm mit den Koordinaten (1,1) aufgerufen.

a) Implementieren Sie die Methode suche_ausgang(int x, int y) rekursiv als Backtracking-Verfahren. Testen Sie Ihre Implementierung, indem Sie den Weg zum Ausgang im Beispiel-Labyrinth versperren. Die Antwort muss dann negativ sein. (10 Punkte)

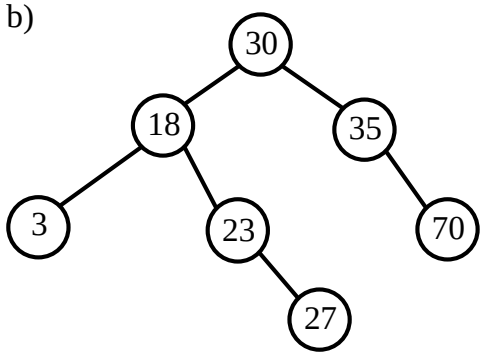
b) Erweitern Sie ihr Programm so, dass ein falscher Weg auf dem Rückweg anders markiert wird (z.b. mit 'f'). Falls ein Ausgang existiert, ist der korrekte Weg so durch die Kreidemarkierung '*' gegeben. Geben Sie das Labyrinth im Falle eines existierenden Wegs aus. (5 Punkte) .

Aufgabe 2: AVL-Bäume erkennen

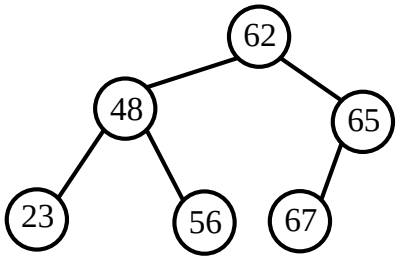
Untersuchen Sie die folgenden Bäume darauf, ob es sich um AVL-Bäume handelt. Für alle Bäume, auf die dies nicht zutrifft, geben Sie bitte eine Begründung an. (5 Punkte)

a)

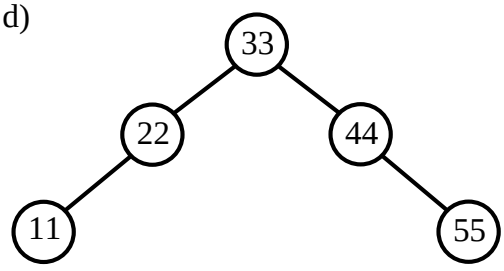
b)



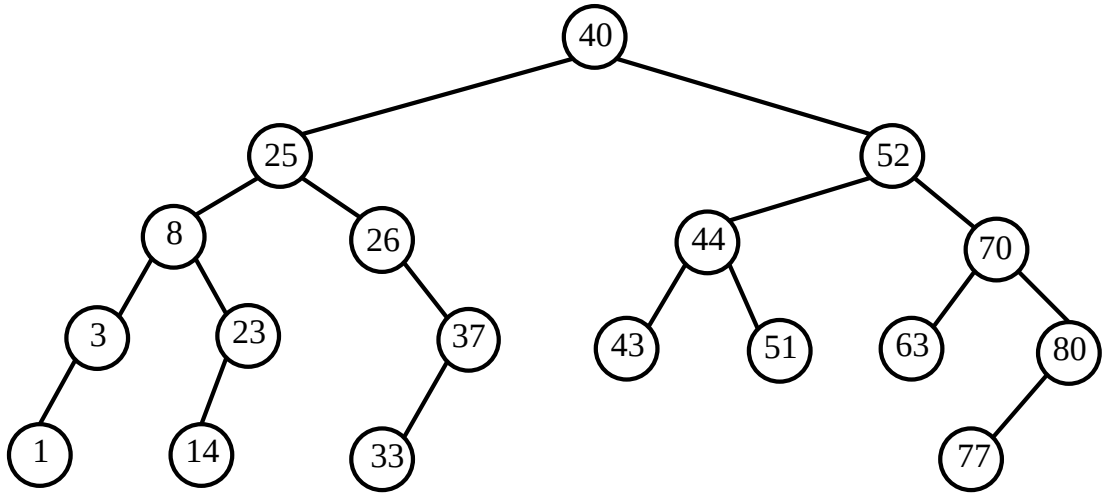
c)



d)



e)



Hinweise

- Lösungen für Programmieraufgaben sind zusätzlich per Email einzureichen.
- Senden Sie Ihre Quelldateien hierzu an die passendende Abgabe-Emailadresse. Die Emailadresse lautet aud?@ifis.uni-luebeck.de, wobei Sie das ? durch Ihre Gruppennummer ersetzen.
- Fügen Sie dem Betreff Ihrer Abgabemail unbedingt die Gruppennummer und die Nummer des Übungsblattes hinzu.
- Papierabgaben ohne Emaileinreichung bzw. Emails ohne Papierabgabe werden nicht gewertet.

Abgabetermin: Donnerstag den 30. April, bis 11 Uhr Abgabekasten IFIS