Übungen 3

Patrick Bucher

10.03.2017

1

1.3

a. a
b. e, e, g, c, k, i
c. b: 3, d: 2, h: 1, j: 1
d. d befindet sich auf dem 2. Niveau
e. auf dem Niveau 3 gibt es 5 Knoten
f. der Knoten j hat die Tiefe 4
g. der Baum hat die Höhe 5
h. der Baum hat die Ordnung 3
i. der Baum ist nicht ausgeglichen, da
    - es Knoten mit 0..3 Kindern gibt
    - es Blätter mit unterschiedlichen Tiefen gibt

2

2.3

a. ein binärer Baum hat, wie jeder andere Baum, eine Wurzel

b. ein binärer Baum hat die Ordnung 2

c. 31 Knoten benötigen im besten Fall 5 Niveaus (31=2^5-1)

    - Anzahl Knoten = Ordnung^(AnzahlNiveaus - 1) - 1
    - Niveau 1: 1 Knoten
    - Niveau 2: 2 Knoten (Total: 3 Knoten)
    - Niveau 3: 4 Knoten (Total: 7 Knoten)
    - Niveau 4: 8 Knoten (Total: 15 Knoten)
    - Niveau 5: 16 Knoten (Total: 31 Knoten)

d. 31 Knoten benötigen im schlechtesten Fall 31 Niveaus

   - auf jedem Niveau ist nur ein Knoten

e. maximale Anzahl Knoten pro Niveau: 2^(Niveau-1)

   1. 1 Knoten
   2. 2 Knoten
   3. 4 Knoten
   4. 8 Knoten
   5. 16 Knoten
   6. 32 Knoten

f. Skizze:

```
  O
 / \
 O O
/
O
```

g. Im Besten Fall ist der binäre Baum geordnet. In diesem Fall kann eine binäre Suche in `n` Elementen mit der Ordnung `O(log n)` angewandt werden. Im schlechtesten Fall ist der Baum so aufgebaut, dass er nur einen Knoten pro Niveau hat. Dann hat die Suche die Ordnung `O(n)`.

3

3.3

a)

a. [Anzahl Elemente] = 2[1] - 1, 15=2^4-1 (vier Niveaus)

```
   .   O
   /       \
   O       O
 /   \   /   \
 O   O   O   O
/ \ / \ / \ / \
O O O O O O O O
```

b.

```
   .   H
   /       \
   D       L
 /   \   /   \
```

[1] Anzahl Niveaus

```
 B   F   J   N
/ \ / \ / \ / \
A C E G I K M O
```

c. Suche nach N, K und O:

   - N: `N > H -> r, N > L -> r, N` gefunden
   - K: `K > H -> r, K < L -> l, K > J -> r, K` gefunden
   - O: `O > H -> r, O > L -> r, O > N -> r, O` gefunden

d.
```
  .   G
  /       \
  B       H
/   \
A   F
   /
   E
\
D
   /
   L
```

e. Dieser “binäre Baum” entspricht einer einfach verketteten Liste

```
A
\
B
\
C
\
D
\
E
\
F
\
G
\
H
```

f. Der Baum muss in der *Inorder-Reihenfolge* traversiert werden, um der Sortierung zu folgen.

g. Algorithmus von der Wurzel aus: [TODO]

4

TODO

5

TODO

6

TODO

7