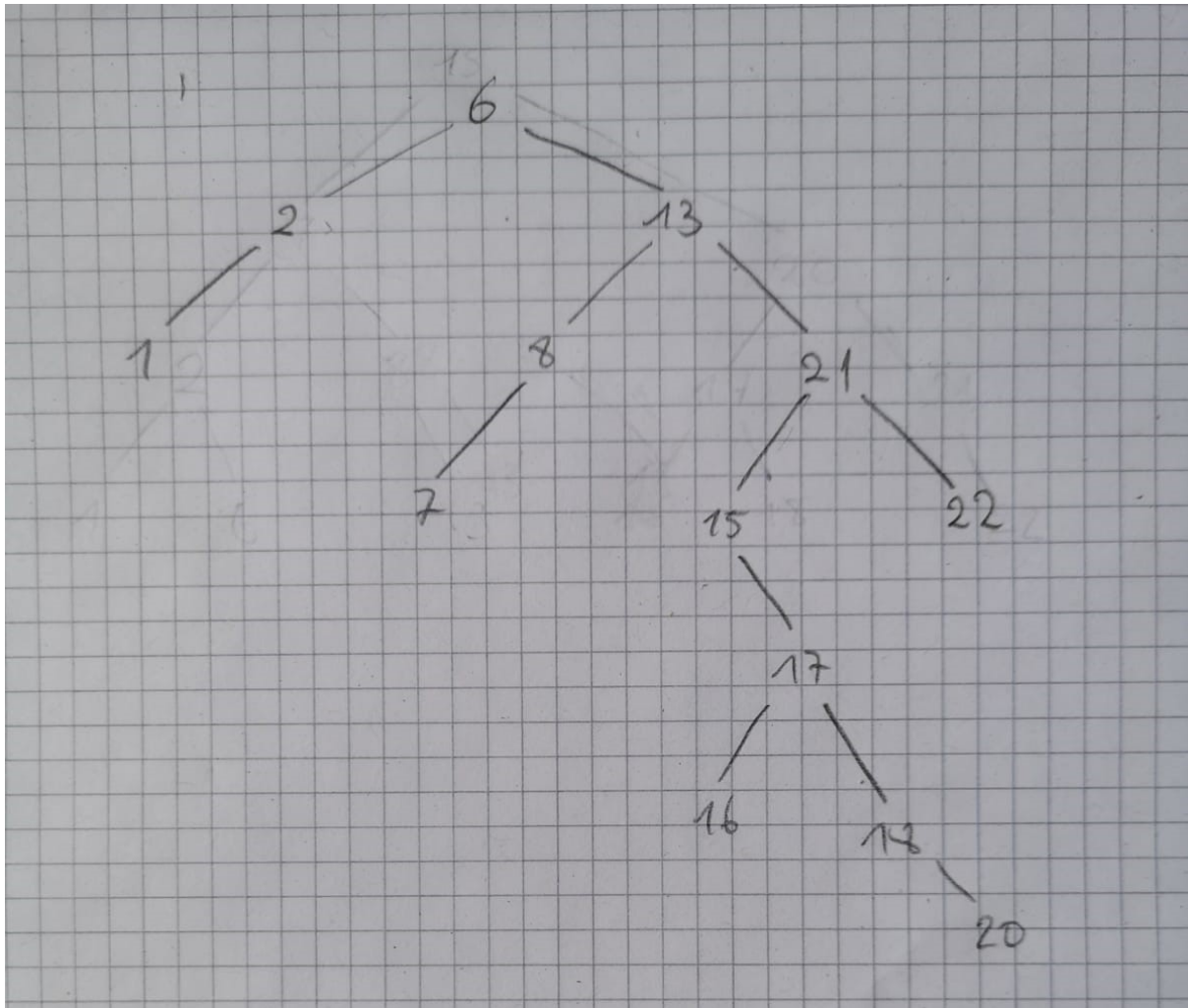


5

Blatt 5

Aufgabe 1

6|13|2|21|8|15|22|7|1|17|16|18|20



sorted:

1|2|6|7|8|13|15|16|17|18|20|21|22

Die Laufzeitkomplexität lautet $O(\log n)$ bei einem ausbalancierten Baum. Bei einem unbalancierten Baum hängt die Suche von der Höhe ab.

Array-Notation:

6	2	13	1	8	21	...	7	15	22	...	17	...	16	18	...	20
---	---	----	---	---	----	-----	---	----	----	-----	----	-----	----	----	-----	----

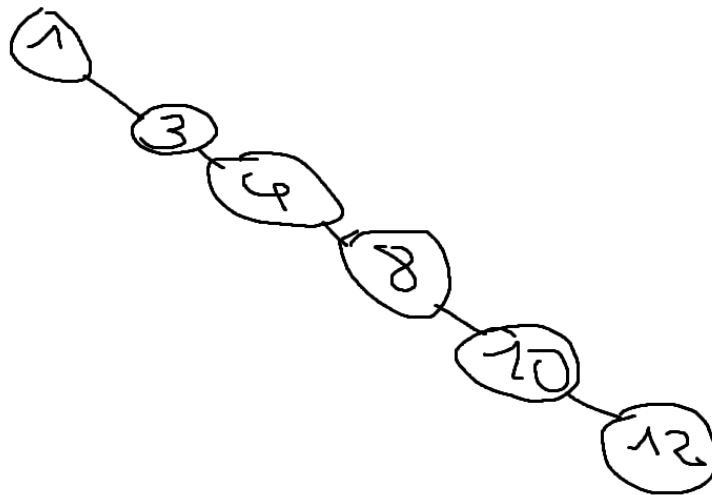
Index:

0	1	2	3	4	5	6	7	...	10	11	12	13	14	15	...	27	28	27	...	56	57	58	59	...	117	118
---	---	---	---	---	---	---	---	-----	----	----	----	----	----	----	-----	----	----	----	-----	----	----	----	----	-----	-----	-----

Aufgabe 2

1. Wenn die gegebene Liste von Keys in geordneter Reihenfolge ist, dann entsteht ein Baum der immer nur einen Zweig folgt. Somit ist es eigentlich wie eine verkettete Liste, weshalb die Laufzeitkomplexität $\Omega(n^2)$ ist.

1 3 4 8 10 12



2. Die Höhe eines Binärbaumes ist die Länge von der Wurzel bis zum tiefsten Blatt. Die minimale Höhe ist wenn jeder Elternteil zwei Kinder hat.

Laut Vorlesung: Ein Binärbaum der Höhe h kann maximal:

$$n = 2^{h+1} - 1$$

Knoten fassen.

Nun kann man nach h die Gleichung umformen:

$$n = 2^{h+1} - 1$$

$$n + 1 = 2^{h+1}$$

$$\log_2(n + 1) = \log_2(2^{h+1})$$

$$\log_2(n + 1) = h + 1$$

$$\log_2(n + 1) - 1 = h$$

Wir sehen, dass die Gleichung übereinstimmt!

3. Zeigen Sie, dass die folgende Aussage wahr ist: Ein Binärbaum mit B Blättern hat mindestens $\log_2(B)$ Zeilen exklusive der Wurzel.

Wir wissen, dass Blätter *externe* Knoten sind und diese hat die Eigenschaft $1 \leq B \leq 2^h$, wenn wir diese Ungleichung nach höchstens h umformen, dann bekommen wir die Mindestanforderung für die Anzahl der Zeilen:

$$1 \leq B \leq 2^h$$

$$\log_2(1) \leq \log_2(B) \leq h * \log_2(2)$$

$$0 \leq \log_2(B) \leq h * 1 \quad \Rightarrow \quad h \geq \log_2(B) \geq 0$$



Beim Logarithmus würde sich das Ungleichzeichen drehen, wenn c in \log_c diese Eigenschaft erfüllt: $0 < c < 1$

Aufgabe 3

```
public boolean hasNext() {  
    return (!queue.isEmpty());  
}
```

```
public int next() {  
    if (queue == null){  
        return 0;  
    }  
    Node current = queue.remove();  
    if (current.left != null) {  
        queue.add(current.left);  
    }  
    if (current.right != null) {  
        queue.add(current.right);  
    }  
    return current.value;  
}
```

Aufgabe 4

Präorder-Traversierung:

O S T E R F E R I E N
1|2|3|4|5|6|7|8|9|10|11

Postorder-Traversierung:

ERTFSIRNEEO

4|5|3|6|2|9|8|11|10|7|1

