

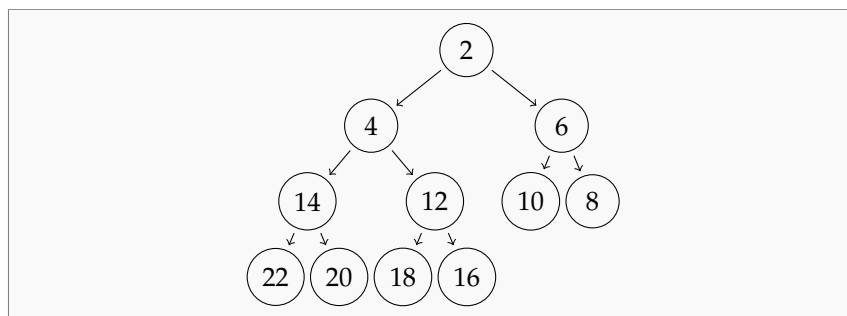
Aufgabe 2

Wir betrachten ein Feld A von ganzen Zahlen mit n Elementen, die über die Indizes $A[0]$ bis $A[n-1]$ angesprochen werden können. In dieses Feld ist ein binärer Baum nach den folgenden Regeln eingebettet: Für das Feldelement mit Index i befindet sich

- der Elternknoten im Feldelement mit Index $\lfloor \frac{i-1}{2} \rfloor$,
- der linke Kindknoten im Feldelement mit Index $2 \cdot i + 1$, und
- der rechte Kindknoten im Feldelement mit Index $2 \cdot i + 2$.

(a) Zeichnen Sie den durch das folgende Feld repräsentierten binären Baum.

i	0	1	2	3	4	5	6	7	8	9	10
A[i]	2	4	6	14	12	10	8	22	20	18	16



(b) Der folgende rekursive Algorithmus sei gegeben:

```

1  procedure magic(i, n : integer) : boolean
2  begin
3    if (i > (n - 2) / 2) then
4      return true;
5    endif
6    if (A[i] <= A[2 * i + 1] and A[i] <= A[2 * i + 2] and
7      magic(2 * i + 1, n) and magic(2 * i + 2, n)) then
8      return true;
9    endif
10   return false;
11 end

12 public class Baum {
13   public static int[] A;
14
15   public static boolean magic(int i, int n) {
16     System.out.println(String.format("i: %s n: %s", i, n));
17     if (i > (n - 2) / 2) {
18       return true;
19     }
20
21     if (A[i] <= A[2 * i + 1] && A[i] <= A[2 * i + 2] && magic(2 * i + 1, n)
22       && magic(2 * i + 2, n)) {
23       return true;
24     }
25   }
26 }
  
```

```

15     return false;
16 }
17
18 public static void main(String[] args) {
19     A = new int[] { 2, 4, 6, 14 };
20     System.out.println(magic(0, 3));
21
22     A = new int[] { 2, 4, 6, 14, 12, 10, 8, 22, 20, 18, 16 };
23     System.out.println(magic(0, 10));
24 }
25 }

```

Gegeben sei folgendes Feld:

i	0	1	2	3
A[i]	2	4	6	14

Führen Sie `magic(0,3)` auf dem Feld aus. Welches Resultat liefert der Algorithmus zurück?

true

- (c) Wie nennt man die Eigenschaft, die der Algorithmus `magic` auf dem Feld A prüft? Wie lässt sich diese Eigenschaft formal beschreiben?

Haldeneigenschaft einer Minhalde.

Die Methode funktioniert jedoch nicht fehlerfrei. `magic(0, 3)` liefert bei dem Feld 2, 4, 6, 14 `true` zurück.

- (d) Welche Ausgaben sind durch den Algorithmus `magic` möglich, wenn das Eingabefeld aufsteigend sortiert ist? Begründen Sie Ihre Antwort.

true. Eine sortierte aufsteigende Zahlenfolge entspricht den Haldeneigenschaften einer Min-Heap.

- (e) Geben Sie zwei dreielementige Zahlenfolgen (bzw. Felder) an, eine für die `magic(0,2)` den Wert `true` liefert und eine, für die `magic(0,2)` den Wert `false` liefert.

- (f) Betrachten Sie folgende Variante `almostmagic` der oben bereits erwähnten Prozedur `magic`, bei der die Anweisungen in Zeilen 3 bis 5 entfernt wurden:

```

1  procedure almostmagic(i, n : integer) : boolean
2  begin
3      // leer
4      // leer
5      // leer
6      if (A[i] <= A[2 * i + 1] and A[i] <= A[2 * i + 2] and
7          magic(2 * i + 1, n) and magic(2 * i + 2, n)) then
8          return true;
9      endif
10     return false;
11 end

```

Beschreiben Sie die Umstände, die auftreten können, wenn `almostmagic` auf einem Feld der Größe `n` aufgerufen wird. Welchen Zweck erfüllt die entfernte bedingte Anweisung?

- (g) Fügen Sie jeweils den angegebenen Wert in den jeweils angegebenen AVL-Baum mit aufsteigender Sortierung ein und zeichnen Sie den resultierenden Baum vor möglicherweise erforderlichen Rotationen. Führen Sie danach bei Bedarf die erforderliche(n) Rotation(en) aus und zeichnen Sie dann den resultierenden Baum. Sollten keine Rotationen erforderlich sein, so geben Sie dies durch einen Text wie "keine Rotationen nötig" an.
- (i) Wert 7 einfügen
 - (ii) Wert 11 einfügen
 - (iii) Wert 5 einfügen