Algorithmen und Datenstrukturen - Hausübung 07

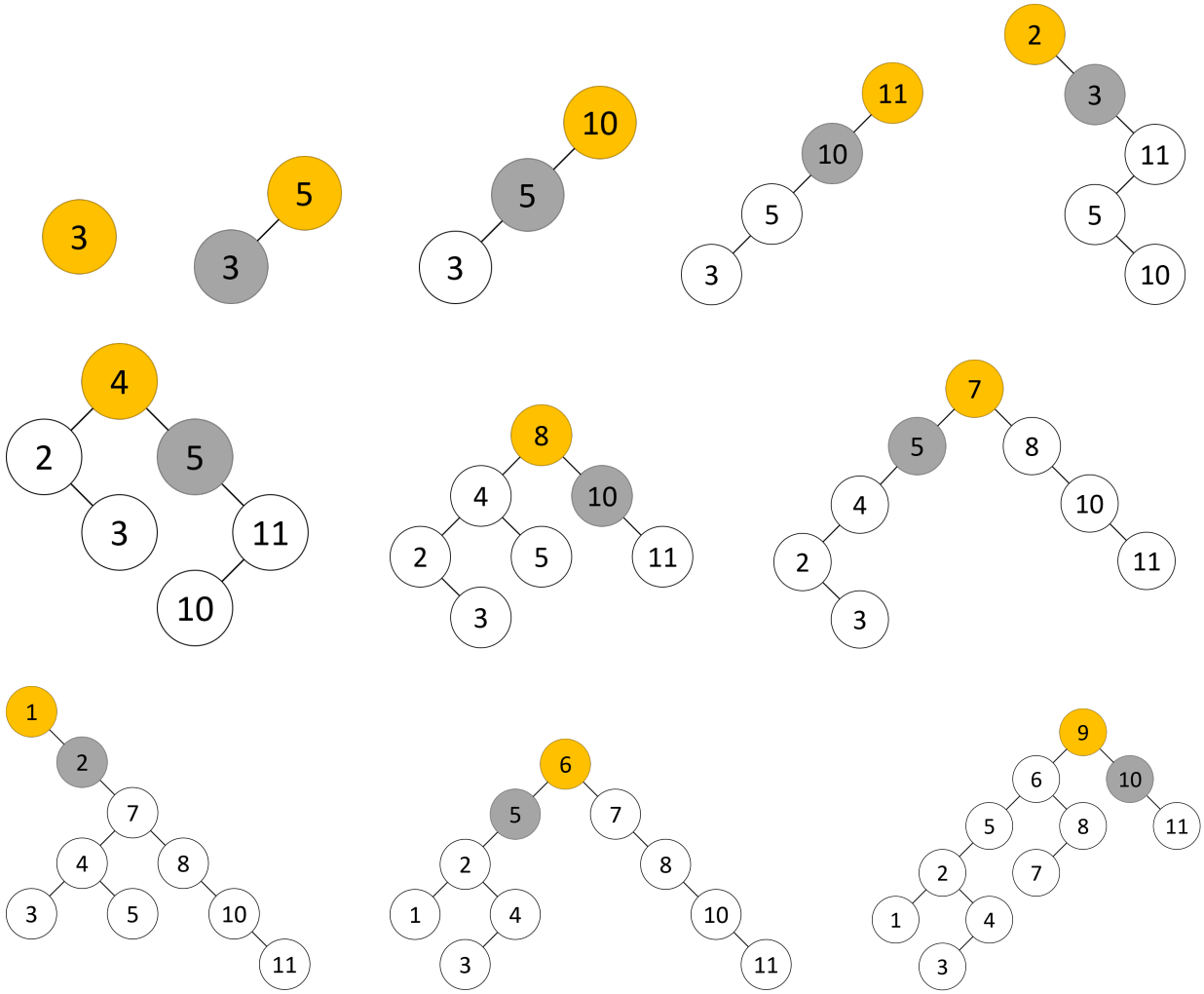
Gruppenmitglieder

- Emre Berber (2957148)
- Christoph Berst (2743394)
- Jan Braun (2768531)

Inhaltsverzeichnis

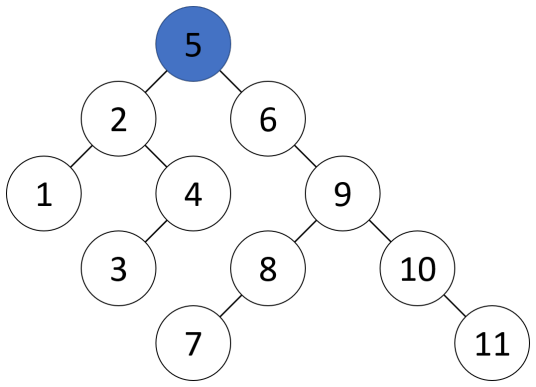
H1

a)

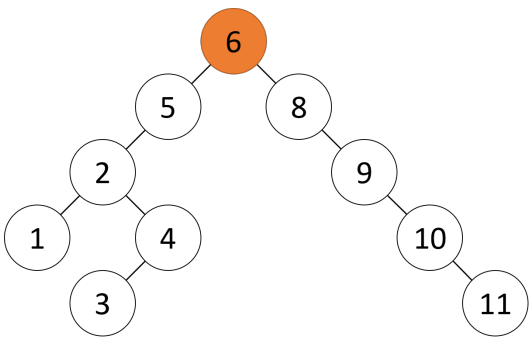


grau: Elterknoten y

b)



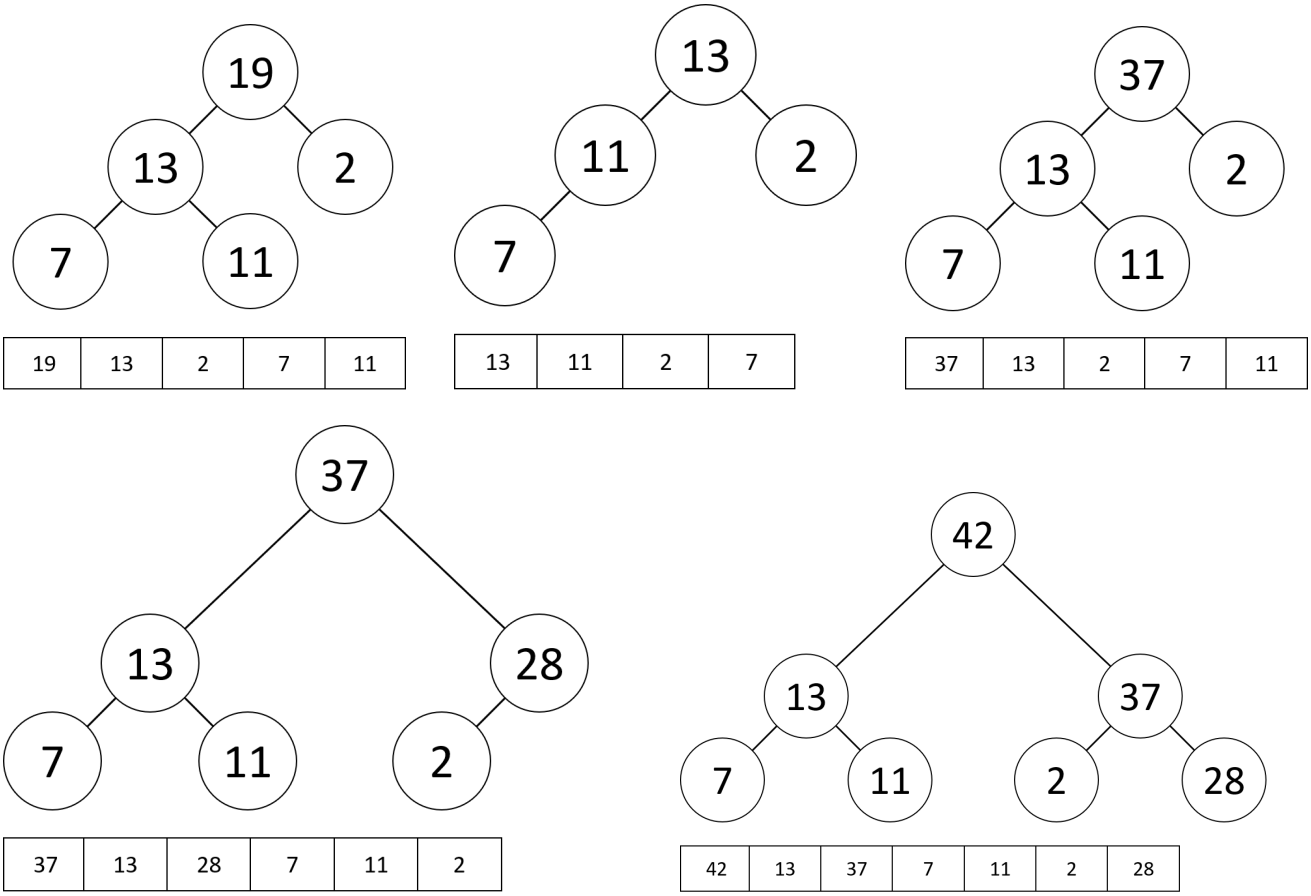
blau: gesuchter Knoten



orange: ”größter” Knoten y aus L

H2

a)



Nach HeapSort(H.A) ist der Baum leer und ein absteigend sortiertes Array ist entstanden.

b)

- BuildHeap(H.A) mit A = {64, 16, 1, 4, 2}
- ExtractMax(H)
- Insert(H,32)
- ExtractMax(H)
- Insert(H,8)

c)

```
delete(H,i)
1   IF isEmpty(H) THEN
2       ERROR 'underflow'
3   IF i == 0 THEN
4       RETURN extractMax(H)
5   IF i == H.size-1 THEN
6       result = H.A
7       H.size = H.size-1
8   ELSE
9       result = H.A
10      H.A = H.A[H.size-1]
11      H.size = H.size-1
12      heapify (H,i)
13  RETURN result
```

H3

a)

```
minPalindrom(w)
1   count = 0
2   FOR i=0 TO w.length − 1
3       j = w.length − 1
4       WHILE true DO
5           WHILE(w[i] ≠ w[j]) DO
6               j--
7           IF i == j THEN
8               BREAK
9           IF checkPalindrom(w, i, j) THEN
10              i = j // next iteration of FOR should be i=j+1
11              BREAK
13      count++
14  RETURN count
```

```
checkPalindrom(w, i, j)
1   WHILE w[i] == w[j] DO
2       i++
3       j--
4   IF i < j THEN
5       RETURN false
6   ELSE
7       RETURN true
```

Mit i iterieren wir von links nach rechts über das Wort w und versuchen dabei jeweils mit Hilfe von j das größtmögliche Palindrom, das mit an i beginnt, zu finden. Wenn wir immer die größtmöglichen Palindrome finden, finden wir gleichzeitig auch die geringste Anzahl an Palindromen, da die Vorherigen den meisten Platz für die Folgenden wegnehmen.

b)

"to be filled in"