

# Algorithmen und Datenstrukturen - Hausübung 07

## Gruppenmitglieder

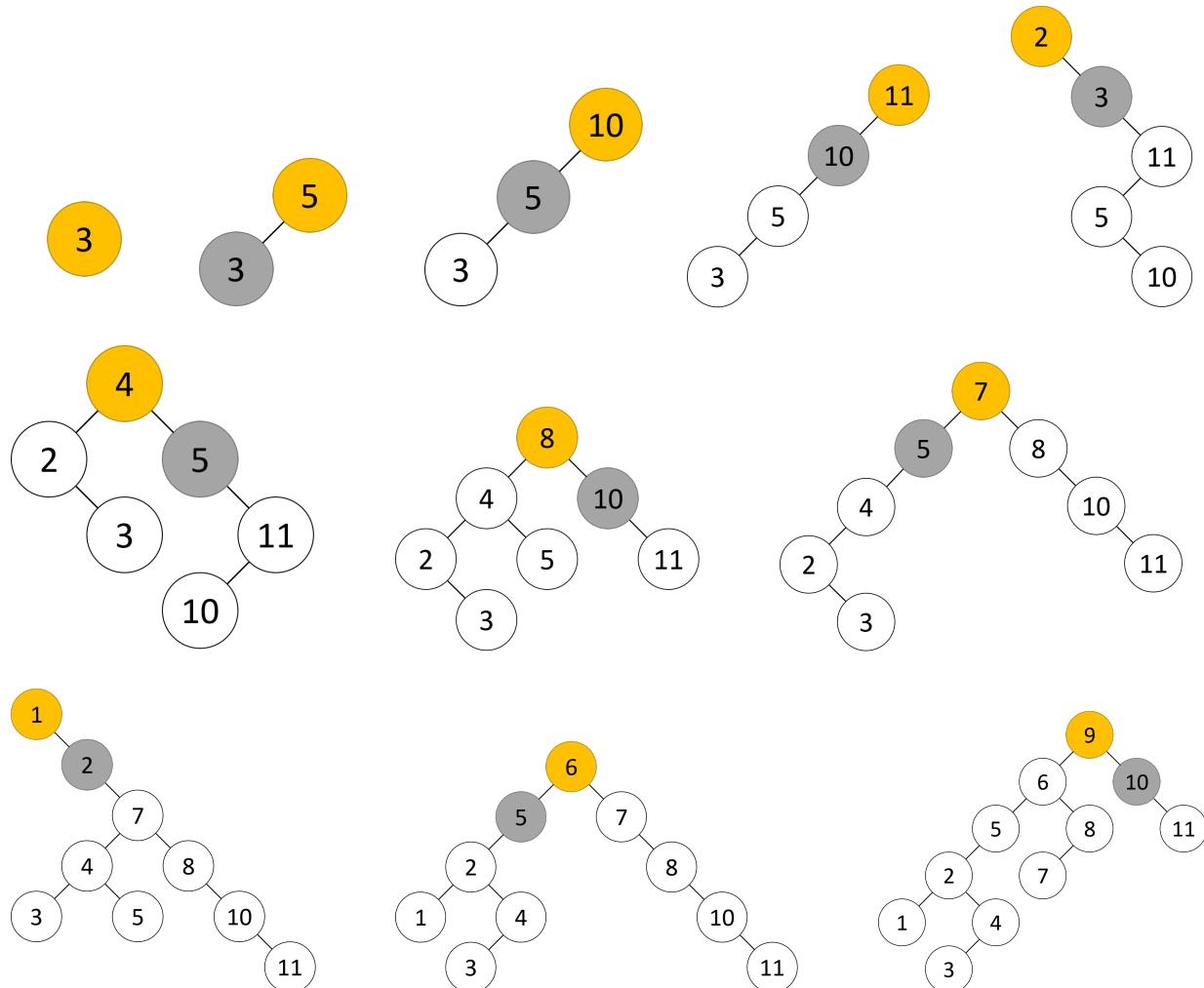
- Emre Berber (2957148)
- Christoph Berst (2743394)
- Jan Braun (2768531)

## Inhaltsverzeichnis

<b>H1</b>		<b>1</b>
a)	.....	1
b)	.....	1
<b>H2</b>		<b>2</b>
a)	.....	2
b)	.....	2
c)	.....	2
<b>H3</b>		<b>2</b>
a)	.....	2
b)	.....	2

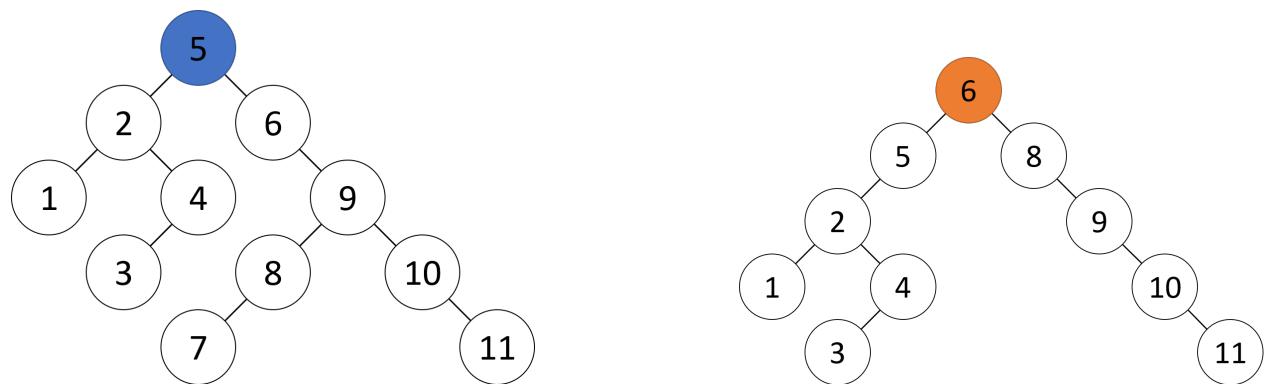
# H1

a)



grau: Elterknoten y

b)



blau: gesuchter Knoten

orange: "größter" Knoten y in L

## H2

a)

"to be filled in"

b)

"to be filled in"

c)

"to be filled in"

## 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 Palidrom, 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"