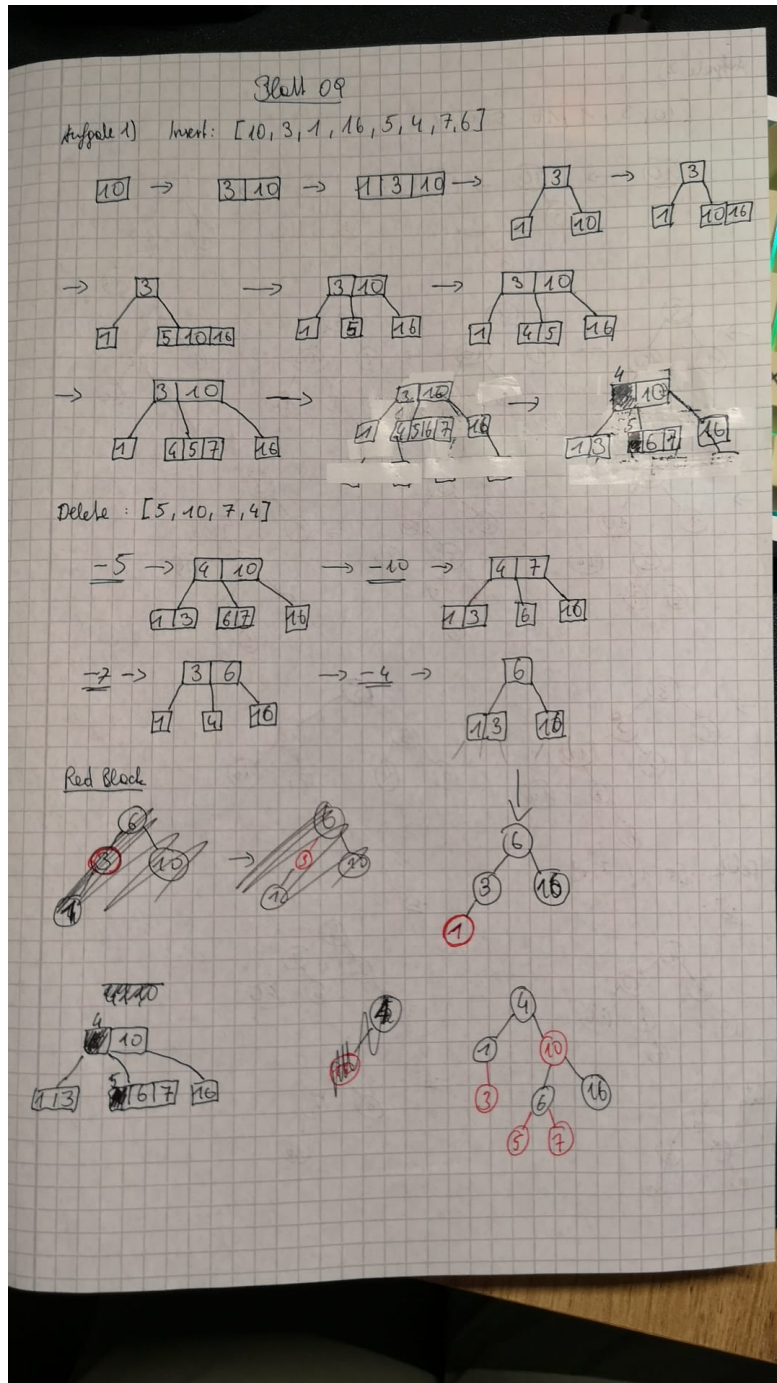


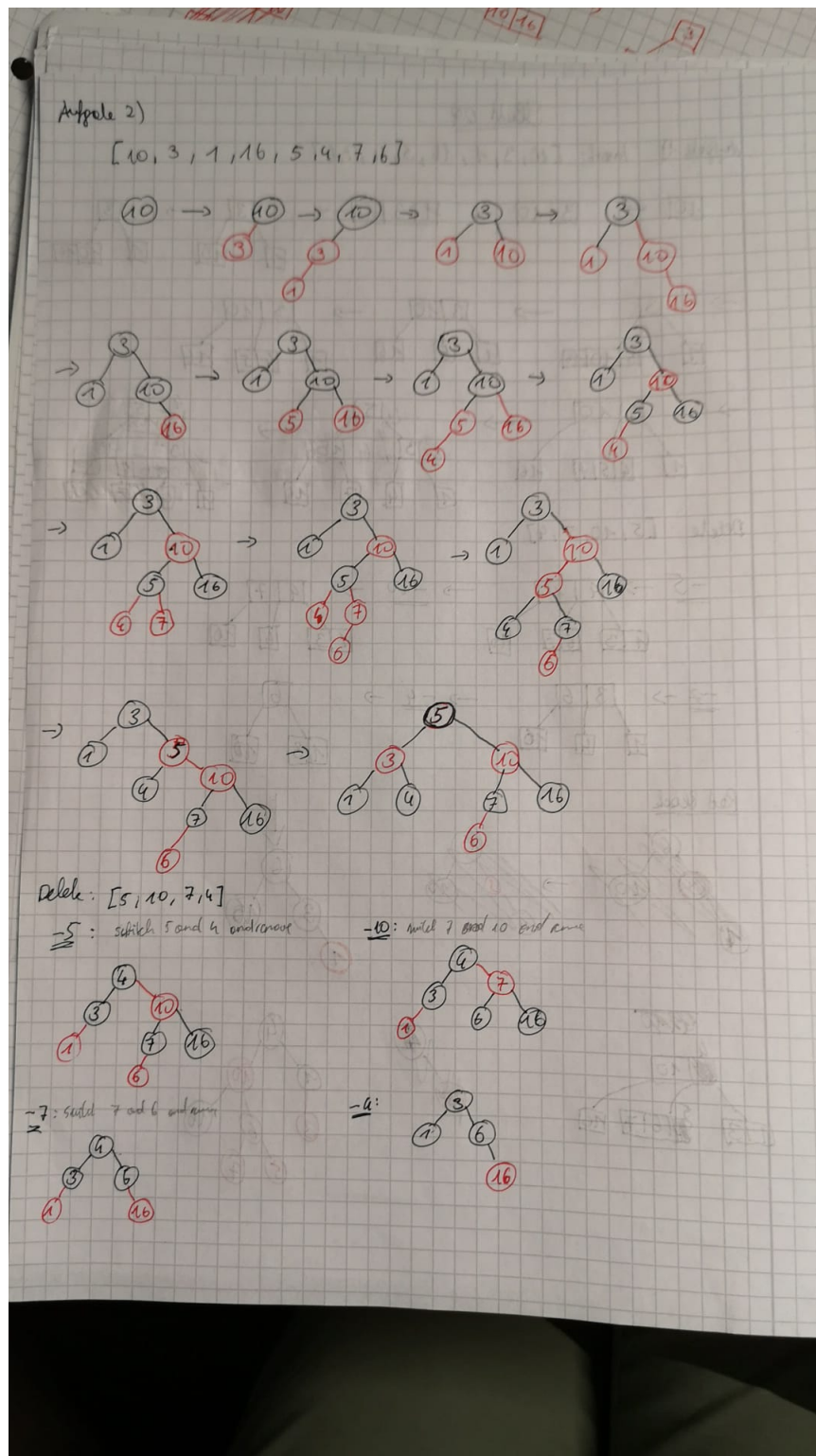


Blatt 09

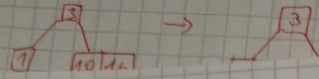
Aufgabe 1



Aufgabe 2



Aufgabe 3

3/10 → 

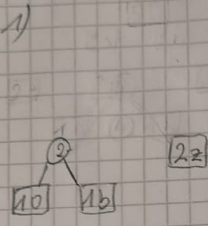
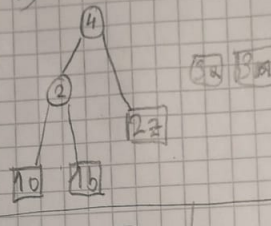
Korrespondierendes (2,4) Baum:

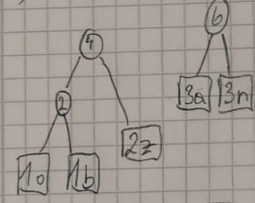
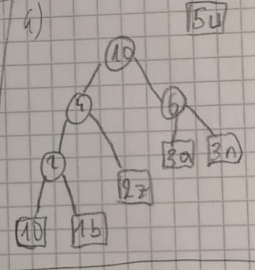
```

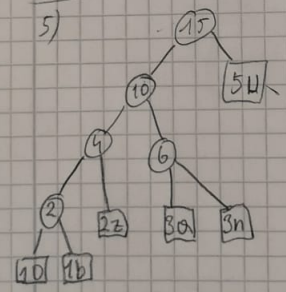
    graph TD
      A[3 5 10] --- B[1]
      A --- C[4]
      A --- D[6 7]
      A --- E[10]
  
```

Aufgabe 3:

Zeichen	u	a	n	z	o	b
Frequenz	5	3	3	2	1	1

1)  

3)  

5) 

b) banana

b: 0001 | a: 010
 n: 011 | a: 010
 n: 011 | a: 010

banana:

0001010011010011010

Aufgabe 4

```
public class CoinExercise {
    public static void coinChange(int S[], int x, int y) {
        int[] availableCoins = S;
        int cost = x;
        int money = y;
        int exchangeMoney = money - cost;

        List<Integer> coins = new ArrayList<>();
        for (int i = S.length; i > 0 ; --i) {
            while (exchangeMoney >= availableCoins[i - 1]) {
                exchangeMoney -= availableCoins[i - 1];
                coins.add(availableCoins[i - 1]);
            }
        }
        System.out.println("your exchange: ");
        coins.forEach(System.out::println);
    }

    public static void main(String[] args) {
        int[] S = {1, 2, 3, 4 , 5, 10};
        int x = 1;
        int y = 10;
        coinChange(S, x, y);
    }
}
```

Nein, dieser Algorithmus gibt unter einem Fall nicht die richtige Lösung zurück und zwar wenn die Liste der Denominatoren nicht aufsteigend sortiert ist.

Beispiel:

```
int[] S = {10, 2, 2, 4, 2, 5 };
int x = 1;
int y = 10;

coinChange(S, x, y);

Output:
5
2
2
```

Hier sehen wir, dass es zwei 2er Münzen der 4 vorzieht. Wie kann man dies umgehen?
Einfach die Liste der Denominatoren sortieren oder sortiert übergeben.