Übungsblatt 8

Adrian Schollmeyer

Aufgabe 1

Aufgabe 2

(a)

$$T(n) = \Theta\left(n^{\log_3 27}\right) \tag{1}$$
$$\text{denn } 3n^2 = O\left(n^{\log_3 27-18}\right) \tag{2}$$

(b)

$$T(n) = \Theta\left(n^{\log_{21}}\right) \tag{3}$$
$$\text{denn } f(n) = O\left(n^{\log_2 1-0{,}0001}\right) \tag{4}$$

(c)

Master-Theorem kann nicht angewendet werden, da $a < 1$.

(d)

$$2n^{2,5}\sqrt{n^3} = 2n^4 \tag{5}$$
$$T(n) = \Theta\left(n^{\log_2 16} \log n\right) \tag{6}$$
$$= \Theta\left(n^4 \log n\right) \tag{7}$$
$$\text{denn } 2n^4 = \Theta\left(n^{\log_2 16}\right) \tag{8}$$

Aufgabe 3

Listing 1: BinarySearchMain.java

```
package me.adrian;

public class BinarySearchMain
{
```

```java
    public static int search(int[] arr, int k)
    {
        return search(arr, k, 0, arr.length);
    }

    public static int search(int[] arr, int k, int i, int j)
    {
        if (j < i)
            throw new IllegalArgumentException("Invalid iterator values");

        int midIndex = (j - i) / 2 + i;
        System.out.println("Checking index " + midIndex);
        int v = arr[midIndex];
        if (i == j - 1) // Last chance to find the object we are looking for ...
            return (k == v) ? midIndex : -1;

        if (k < v) {
            return search(arr, k, i, midIndex);
        } else if (k > v) {
            return search(arr, k, midIndex, j);
        } else if (k == v) {
            return midIndex;
        }
        return -1;
    }

    public static void main(String[] args)
    {
        int[] a = { 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16 };
        System.out.println(search(a, 8));
        System.out.println(search(a, 3));
    }
}
```

Aufgabe 4

(a)

24	**17**	13	42	9	37	1
17	24	**13**	42	9	37	1
13	17	24	**42**	9	37	1
13	17	24	42	**9**	37	1
9	13	17	24	42	**37**	1
9	13	17	24	37	42	**1**
1	9	13	17	24	37	42

(b)

24	17	13	**42**	9	37	1
24	17	13	1	9	**37**	42
24	17	13	1	9	37	42
9	**17**	13	1	24	37	42
9	1	**13**	17	24	37	42
9	1	13	17	24	37	42
1	9	13	17	24	37	42

(c)

24	**17**	13	42	9	37	1
17	**24**	**13**	42	9	37	1
17	**13**	24	42	9	37	1
13	17	24	**42**	**9**	37	1
13	17	**24**	**9**	42	37	1
13	**17**	**9**	24	42	37	1
13	**9**	17	24	42	37	1
9	13	17	24	**42**	**37**	1
9	13	17	24	37	**42**	**1**
9	13	17	24	**37**	**1**	42
9	13	17	**24**	**1**	37	42
9	13	**17**	**1**	24	37	42
9	**13**	**1**	17	24	37	42
9	**1**	13	17	24	37	42
1	9	13	17	24	37	42

(d)

Wird aufgrund des Fortschritts in der Vorlesung ausgelassen.

(e)

Wird aufgrund des Fortschritts in der Vorlesung ausgelassen.

Aufgabe 5

Listing 2: BubbleSortMain.java

```
package me.adrian;

public class BubbleSortMain
{
    private static String atos(int[] a)
    {
```

```
        StringBuffer sb = new StringBuffer("{");
        for (int i = 0; i < a.length; i++) {
            if (i >= a.length - 1) {
                sb.append(a[i] + "}");
            } else {
                sb.append(a[i] + ", ");
            }
        }
        return sb.toString();
    }

    private static void swap(int[] a, int i, int j)
    {
        if (i == j)
            return;

        int iVal = a[i];
        a[i] = a[j];
        a[j] = iVal;
    }

    public static void sort(int[] a)
    {
        for (int i = a.length; i > 0; i--) {
            int max = Integer.MIN_VALUE;
            int maxIndex = -1;

            for (int j = 0; j < i; j++) {
                if (a[j] > max) {
                    max = a[j];
                    maxIndex = j;
                }
            }

            swap(a, i - 1, maxIndex);
        }
    }

    public static void main(String[] args)
    {
        int[] arr = { -12, 3, 4, 6, 344, 4, 3, 2, 2, 34, 554, -3, -23234,
            -245777654, 12124, 2222205 };
        sort(arr);
        System.out.println(atos(arr));
    }
}
```