

Algorithmen und Datenstrukturen
SoSe 2024

Übungsblatt 1: Grundlagen

Tutorien: 22.04-26.04.2024

Aufgabe 1-1 *Basics zu Algorithmen*

Geben Sie bei folgenden Algorithmen an, welches Problem sie lösen.

Bestimmen Sie außerdem, welche grundlegenden Eigenschaften erfüllt sind (Allgemeinheit, Determiniertheit, Determinismus, Terminierung, Effizienz).

- (a) `public static boolean f0(int a) {
 if (a == 0)
 return true;
 if (a == 1)
 return false;
 if (a < 0)
 return f0(-a);
 return f0(a - 2);
}`
- (b) `public static boolean f1(String str1, String str2) {
 char[] array1 = new char[Character.MAX_VALUE];
 char[] array2 = new char[Character.MAX_VALUE];
 for(char c : str1.toCharArray()) {
 array1[c]++;
 }
 for(char c : str2.toCharArray()) {
 array2[c]++;
 }
 for(char c = 0; c < Character.MAX_VALUE; c++) {
 if(array1[c] != array2[c])
 return false;
 }
 return true;
}`
- (c) `public static double f2(double a, int n) {
 double x = 1.0;
 for (int i = 0; i < n; i++)
 x = 0.5 * (x + a / x);
 return x;
}`
- (d) `public static double f3(int n) {
 double sum = 0.0;
 for (int i = 0; i < n; i++) {`

```

        double x = Math.random();
        double y = Math.random();
        if(Math.sqrt(x*x+y*y) <= 1.0)
            sum++;
    }
    return 4.0*sum/n;
}

```

```

(e) public static int f4(int a, int b) {
    if (b == 1)
        return a;
    return a * f4(a, b - 1);
}

```

```

(f) public static int f5(int a, int b) {
    if (a == 0)
        return b + 1;
    if (b == 0)
        return f5(a - 1, 1);
    return f5(a - 1, f5(a, b - 1));
}

```

```

(g) public static void f6() {
    LinkedList<Integer> list = new LinkedList<Integer>();
    int n = 2;
    list.add(n);
    n = 3;
    list.add(n);
    while (true) {
        n += 2;
        boolean add = true;
        for (Integer m : list) {
            if (n % m == 0)
                add = false;
        }
        if (add) {
            list.add(n);
            System.out.println(n);
        }
    }
}

```

Aufgabe 1-2 *Spaß mit Determiniertheit und Determinismus*

Um den Unterschied zwischen Determiniertheit und Determinismus zu verstehen, betrachten wir folgende Funktion, die zwar determiniert aber nicht deterministisch ist. Begründen Sie warum das so ist.

```

public static int f7() {
    int a = 0;
    int b = 0;
    if (Math.random() < 0.5) {
        a++;
        b++;
    } else {
        b++;
        a++;
    }
}

```

```
}  
  
return a + b;  
}
```