

Algorithmen und Datenstrukturen
SS 2021

Übungsblatt 1: Grundlagen

Tutorien: 26.04.-30.04.2021

Hinweis zum Übungsablauf:

Übungsblätter erscheinen wöchentlich auf Uni2Work. Nach ihrer Veröffentlichung gibt es eine Woche Zeit, sich mit den Inhalten zu beschäftigen und im besten Fall die Aufgaben eigenständig zu lösen. In der darauf folgenden Woche gibt es die Möglichkeit, sich von uns angefertigte Lösungsvideos anzusehen, die wenigstens einen möglichen Lösungsweg aufzeigen. Viele Aufgaben können auf unterschiedliche Arten gelöst werden und wir erheben keinen Anspruch auf absolute Vollständigkeit. Fragen, die dann noch offen sind oder die sich durch die unterschiedlichen Lösungsvarianten ergeben, können anschließend in den Fragestunden mit den Tutoren geklärt werden. Bereiten Sie sich bitte auf diese Einheiten aktiv vor, in dem Sie sich zumindest die Aufgaben vorher ansehen. Unsere Tutoren werden auf Ihren Input reagieren, doch wenn keine Diskussionspunkte von Ihnen angetragen werden, können die Fragestunden sehr kurz werden. Erwarten Sie bitte nicht, dass Mitstudenten schon alle relevanten Fragen für Sie stellen oder dass unsere Tutoren ein Programm vorbereiten, um die Stille zu füllen. Wir regen auch an, die Fragen im Vorfeld bereits an die Tutoren zu senden. Je nach Komplexität einer Frage kann diese nicht aus dem Stegreif beantwortet werden und wird ggfs. auf die nächste Fragestunde verschoben. Das rechtzeitige Einsenden von Fragen hilft da ungemein.

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;  
}
```

```

        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;
    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 Zusatzaufgabe: 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:

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

Die Funktion wird immer $1 + 1 = 2$ zurückgeben, ist also determiniert (gleiche Eingabe führt zu gleicher Ausgabe).

Die Zustände, die bei der Abarbeitung durchlaufen werden, sind aber vom Ergebnis von *Math.random()* abhängig, das zufällig ist. Die Abarbeitungsfolge ist also nicht eindeutig bestimmt. Es kann sein, dass der *if*-Block oder der *else*-Block ausgeführt wird. Daher ist die Funktion nicht deterministisch.

Aufgabe 1-3 Zusatz-Zusatz-Aufgabe: Einloggen zur Fragestunde

Richten Sie sich ein Profil bei Discord ein. Loggen Sie sich pünktlich zu Ihrer Fragestunde ein und betreten Sie den Sprachchannel. Entmuten Sie sich und sagen Sie Ihrem Tutor Hallo. Anschließend können Sie sich gerne wieder muten.

Hinweis: Wir beobachten oft, dass Nutzer mehrere Wochen in den Sprachkanälen stumm verbleiben. Wenn Sie sich wundern, warum Ihr Username in einem der Sprachkanäle verbleibt, dann versuchen Sie doch einfach, den Kanal zu verlassen, wenn keine Fragestunden stattfinden. Der Button dazu ist unten links zu finden und in folgendem Bild markiert. Auch wenn Ihre Sprachverbindung deaktiviert ist, können Sie alle Textnachrichten auf dem Server weiterhin lesen und schreiben. Wenn keine Veranstaltungen stattfinden, dürfen Sie natürlich jederzeit die Sprachkanäle für beliebige (aber juristisch unbedenkliche) Tätigkeiten nutzen.

