



[Zur LVA in TISS](#)

[Dashboard](#) / [Meine Kurse](#) / [185.A91-2021W](#) / [TUWEL-Tests](#) / [2. TUWEL-Test \(20.01. A\)](#)

Begonnen am Donnerstag, 20. Januar 2022, 18:22

Status Beendet

Beendet am Donnerstag, 20. Januar 2022, 19:21

Verbrauchte Zeit 59 Minuten 33 Sekunden

Bewertung 15,00 von 20,00 (75%)

Sie haben mehrere Code-Fragmente gegeben. Jedes Code-Fragment beinhaltet einen Kommentar, der den Inhalt des Arrays `data` nach der Ausführung des Code-Fragments beschreiben soll. Beurteilen Sie die Richtigkeit der Kommentare für beliebige Werte von $n > 1$!

Hinweis: Mit $[a, b]$ wird das Intervall von a (inklusive) bis b (inklusive) beschrieben.

Richtig Falsch

```
int[] data = new int[n];
int i = n / 2;
while (data[i] == 0) {
    data[i] = i + 1;
    i = (i + 1) % n;
}
// data enthält nur Werte != 0
```



```
int[] data = new int[n];
int i = 0;
while (i < data.length / 2) {
    data[i] = 1;
    data[data.length - 1 - i] = data[i];
    i++;
}
// data enthält nur Werte > 0
```



```
int[] data = new int[n];
int i = 0;
while (i != data.length) {
    data[i] = i++;
}
// data enthält nur Werte aus dem Intervall [1, n - 1]
```



```
int[] data = new int[n];
data[0] = 1;
int i = 1;
while (i < data.length) {
    data[i] = data[i - 1];
    i++;
}
// data enthält nur Werte > 1
```



```
int[] data = new int[n];
int i = n / 2;
while (data[i] == 0) {
    data[i] = i + 1;
    i = (i + 1) % n;
}
// data enthält nur Werte != 0
```

: Richtig

```
int[] data = new int[n];
int i = 0;
while (i < data.length / 2) {
    data[i] = 1;
    data[data.length - 1 - i] = data[i];
    i++;
}
// data enthält nur Werte > 0
```

: Falsch

```
int[] data = new int[n];
int i = 0;
while (i != data.length) {
    data[i] = i++;
}
// data enthält nur Werte aus dem Intervall [1, n - 1]
```

: Falsch



```

int i = 1;
while (i < data.length) {
    data[i] = data[i - 1];
    i++;
}
// data enthält nur Werte > 1

```

: Falsch

Frage 2

Richtig

Erreichte Punkte 2,00 von 2,00

Sie haben folgende Methode gegeben:

```

private static int[] sumSlices(int[][] input, int col1, int col2) {
    int[] r = new int[input.length];
    for (int i = 0; i < input.length; i++) {
        int a = 0;
        int b = 0;
        if (col1 < input[i].length) { // Zeile A
            a = input[i][col1];
        }
        if (col2 < input[i].length) { // Zeile B
            b = input[i][col2];
        }
        r[i] = a + b;
    }
    return r;
}

```

Gehen Sie davon aus, dass als Vorbedingung `input != null`, `input[i] != null` für alle gültigen `i`, `col1 >= 0` und `col2 >= 0` gilt.

Beurteilen Sie die Richtigkeit der folgenden Aussagen!

Richtig	Falsch		
<input type="radio"/>	<input checked="" type="radio"/>	Enthält <code>input</code> nicht den Wert 0, dann kann eine 0 auch nicht im Ergebnisarray vorkommen.	✓
<input checked="" type="radio"/>	<input type="radio"/>	Ohne die if-Bedingung in Zeile A könnte es trotz Vorbedingungen zu einer Ausnahme kommen.	✓
<input checked="" type="radio"/>	<input type="radio"/>	Ohne die if-Bedingung in Zeile B könnte es trotz Vorbedingungen zu einer Ausnahme kommen.	✓
<input type="radio"/>	<input checked="" type="radio"/>	Die Methode erzeugt ein Array, bei dem alle Elemente größer 0 sind.	✓

Enthält `input` nicht den Wert 0, dann kann eine 0 auch nicht im Ergebnisarray vorkommen.: Falsch
 Ohne die if-Bedingung in Zeile A könnte es trotz Vorbedingungen zu einer Ausnahme kommen.: Richtig
 Ohne die if-Bedingung in Zeile B könnte es trotz Vorbedingungen zu einer Ausnahme kommen.: Richtig
 Die Methode erzeugt ein Array, bei dem alle Elemente größer 0 sind.: Falsch

Die Methode `add(int a, int b)` gibt das Ergebnis $a + b$ zurück.

Einige Beispiele für korrektes Verhalten:

`add(2, 4)` liefert 6 zurück
`add(3, 2)` liefert 5 zurück
`add(0, 2)` liefert 2 zurück

Die Vorbedingung lautet: $a \geq 0, b \geq 0$.

Sie haben unterschiedliche rekursive Implementierungen gegeben. Beurteilen Sie die Richtigkeit dieser Implementierungen!

Richtig Falsch

❌

✅

```
private static long add(int a, int b) {
    if (a == 0 || b == 0) {
        return 0;
    } else {
        return add(a - 1, b) + 1;
    }
}
```



❌

✅

```
private static long add(int a, int b) {
    if (a <= 0) {
        return b;
    } else if (b <= 0) {
        return a;
    } else {
        return add(a, b - a) + a;
    }
}
```



✅

❌

```
private static long add(int a, int b) {
    if (a == 0) {
        return b;
    } else if (b == 0) {
        return a;
    } else if (a < b) {
        return add(a - 1, b) + 1;
    } else {
        return add(a, b - 1) + 1;
    }
}
```



✅

❌

```
private static long add(int a, int b) {
    if (a == 0) {
        return b;
    } else if (b == 0) {
        return a;
    } else {
        return add(a, b - 1) + 1;
    }
}
```



```
private static long add(int a, int b) {
    if (a == 0 || b == 0) {
        return 0;
    } else {
        return add(a - 1, b) + 1;
    }
}
```

: Falsch



```

    return b;
} else if (b <= 0) {
    return a;
} else {
    return add(a, b - a) + a;
}
}

```

: Falsch

```

private static long add(int a, int b) {
    if (a == 0) {
        return b;
    } else if (b == 0) {
        return a;
    } else if (a < b) {
        return add(a - 1, b) + 1;
    } else {
        return add(a, b - 1) + 1;
    }
}

```

: Richtig

```

private static long add(int a, int b) {
    if (a == 0) {
        return b;
    } else if (b == 0) {
        return a;
    } else {
        return add(a, b - 1) + 1;
    }
}

```

: Richtig

Frage 4

Teilweise richtig

Erreichte Punkte 1,00 von 2,00

Sie haben folgende Methode gegeben:

```

private static String mix(String a, String b) {
    if (a.isEmpty() && b.isEmpty()) return ""; // Zeile A
    if (a.isEmpty()) return mix("#", b);
    if (b.isEmpty()) return mix(a, "#");
    return "" + a.charAt(0) + b.charAt(0) + mix(a.substring(1), b.substring(1)); // Zeile B
}

```

Beurteilen Sie die Richtigkeit der folgenden Aussagen!

Richtig	Falsch		
<input checked="" type="radio"/>	<input type="radio"/>	Die Methode kann für bestimmte Argumente den String <code>a###bdce</code> als Rückgabewert erzeugen.	✗
<input type="radio"/>	<input checked="" type="radio"/>	Wird in Zeile A der Oder-Operator () verwendet, dann kann es zu einer <code>StringIndexOutOfBoundsException</code> kommen.	✓
<input checked="" type="radio"/>	<input type="radio"/>	In Zeile A befindet sich der Basisfall.	✓
<input type="radio"/>	<input checked="" type="radio"/>	Wird Zeile B in <code>return "" + b.charAt(0) + a.charAt(0) + mix(b.substring(1), a.substring(1));</code> geändert, dann wird dieselbe Ausgabe erzeugt, wie in der oben gegebenen Version.	✓

Die Methode kann für bestimmte Argumente den String `a###bdce` als Rückgabewert erzeugen.: Richtig

Wird in Zeile A der Oder-Operator (||) verwendet, dann kann es zu einer `StringIndexOutOfBoundsException` kommen.: Falsch

In Zeile A befindet sich der Basisfall.: Richtig

Wird Zeile B in `return "" + b.charAt(0) + a.charAt(0) + mix(b.substring(1), a.substring(1));` geändert, dann wird dieselbe Ausgabe erzeugt, wie in der oben gegebenen Version.: Falsch

Sie haben folgende Methode gegeben:

```
private static void generate(String text, int n) {  
    if (n >= 0) {  
        generate(text + "A", n - 1);  
        System.out.println(text);  
        generate(text + "B", n - 1);  
    }  
}
```

Beurteilen Sie die Richtigkeit der folgenden Aussagen.

Richtig	Falsch		
<input checked="" type="radio"/>	<input type="radio"/>	Bei dieser Methode ist der Basisfall implizit vorhanden (falls der if-Zweig nicht betreten wird).	✓
<input type="radio"/>	<input checked="" type="radio"/>	Die Methode generiert genau 2^n Ausgaben (Zeilen).	✓
<input checked="" type="radio"/>	<input type="radio"/>	Die Methode kann auch nur <input type="text" value="B"/> als einzige Ausgabe erzeugen.	✓
<input checked="" type="radio"/>	<input type="radio"/>	Die Methode kann auch nur <input type="text" value="A"/> als einzige Ausgabe erzeugen.	✓

Bei dieser Methode ist der Basisfall implizit vorhanden (falls der if-Zweig nicht betreten wird).: Richtig
Die Methode generiert genau 2^n Ausgaben (Zeilen). : Falsch
Die Methode kann auch nur als einzige Ausgabe erzeugen.: Richtig
Die Methode kann auch nur als einzige Ausgabe erzeugen.: Richtig



Sie haben mehrere Code-Fragmente mit dazugehörigen Kommentaren gegeben. Jeder Kommentar beschreibt die kleinste obere Schranke für die Laufzeit in Abhängigkeit von n . Dabei steht " n^x " für n^x . Beurteilen Sie die Richtigkeit der Kommentare!

Richtig Falsch

```
// Die Laufzeit liegt in O(n)
int sum = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j += n / 2) {
        sum += i + j;
    }
}
```



```
// Die Laufzeit liegt in O(n)
int sum = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n / 2; j += 4) {
        sum += i + j;
    }
}
```



```
// Die Laufzeit liegt in O(1)
int sum = 0;
for (int i = 0; i < 10; i++) {
    for (int j = 0; j < 10; j++) {
        sum += i + j;
    }
}
```



```
// Die Laufzeit liegt in O(n^3)
int sum = 0;
for (int i = 0; i < n * n; i++) {
    for (int j = 0; j < n / 2; j++) {
        sum += i + j;
    }
}
```



```
// Die Laufzeit liegt in O(n)
int sum = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j += n / 2) { : Richtig
        sum += i + j;
    }
}
```

```
// Die Laufzeit liegt in O(n)
int sum = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n / 2; j += 4) { : Falsch
        sum += i + j;
    }
}
```

```
// Die Laufzeit liegt in O(1)
int sum = 0;
for (int i = 0; i < 10; i++) {
    for (int j = 0; j < 10; j++) { : Richtig
        sum += i + j;
    }
}
```



```

for (int i = 0; i < n * n; i++) {
    for (int j = 0; j < n / 2; j++) {
        sum += i + j;
    }
}

```

: Richtig

Frage 7

Teilweise richtig

Erreichte Punkte 1,00 von 2,00

Gegeben sei die folgende Variante des linearen Suchalgorithmus.

```

private static final int NOT_FOUND = -1;

private static int search(int[] data, int key) {
    int i = 0;
    int j = data.length - 1;
    while (i <= j) {
        if (data[i] == key) {
            return i;
        } else if (data[j] == key) {
            return j;
        }
        i++;
        j--;
    }
    return NOT_FOUND;
}

```

Vorbedingung: data != null und data.length >= 0

Beurteilen Sie die Richtigkeit der folgenden Aussagen zu dieser Variante!

Richtig	Falsch		
<input type="radio"/>	<input checked="" type="radio"/>	Die Laufzeit im Worst-Case liegt in $O(\log n)$, wobei n der Länge von <code>data</code> entspricht.	✗
<input type="radio"/>	<input checked="" type="radio"/>	Befindet sich der gesuchte Wert <code>key</code> an der ersten Stelle im Array, dann wird -1 zurückgeliefert.	✓
<input type="radio"/>	<input checked="" type="radio"/>	Hat <code>data</code> eine ungerade Länge, dann kann das Element in der Mitte des Arrays nicht gefunden werden.	✓
<input checked="" type="radio"/>	<input type="radio"/>	Kommt der gesuchte Wert <code>key</code> in <code>data</code> mehrmals vor, dann kann auch der größte Index, an dem der Wert vorkommt, zurückgeliefert werden.	✓

Die Laufzeit im Worst-Case liegt in $O(\log n)$, wobei n der Länge von `data` entspricht.

: Falsch

Befindet sich der gesuchte Wert `key` an der ersten Stelle im Array, dann wird -1 zurückgeliefert.: Falsch

Hat `data` eine ungerade Länge, dann kann das Element in der Mitte des Arrays nicht gefunden werden.: Falsch

Kommt der gesuchte Wert `key` in `data` mehrmals vor, dann kann auch der größte Index, an dem der Wert vorkommt, zurückgeliefert werden.: Richtig

Gegeben sei der aus den Vorlesungsfolien bekannte Sortieralgorithmus Selectionsort.

```
private static void selectionSort(int[] data) {
    for (int i = 0; i < data.length - 1; i++) {
        int min = i;
        for (int j = i + 1; j < data.length; j++) {
            if (data[j] < data[min]) {
                min = j;
            }
        }
        exchange(data, i, min);
        // Zeile A
    }
}

private static void exchange(int[] a, int i, int j) {
    int swap = a[i];
    a[i] = a[j];
    a[j] = swap;
}
```

Sie müssen überprüfen, ob für eine Anfangsbelegung von `data` die gegebene mögliche Belegung von `data` nach irgendeinem Durchlauf der inneren Schleife in Zeile A auftreten kann. Beispiel:

- Anfangsbelegung von `data`: [8, 4, 6, 5, 2, 1]
- Mögliche Belegung: [1, 2, 6, 5, 4, 8] (nach 2 Durchläufen der inneren Schleife)
- Nicht möglich: [1, 2, 4, 6, 5, 8]

Beurteilen Sie die Richtigkeit der folgenden Aussagen zu den Belegungen für unterschiedliche Ausgangsbelegungen von `data`!

Richtig	Falsch		
<input type="radio"/>	<input checked="" type="radio"/>	<ul style="list-style-type: none"> • Ausgangsbelegung: [3, 6, 8, 2, 5, 4] • Mögliche Belegung: [2, 3, 4, 6, 8, 5] 	✓
<input type="radio"/>	<input checked="" type="radio"/>	<ul style="list-style-type: none"> • Ausgangsbelegung: [9, 7, 6, 5, 3, 1] • Mögliche Belegung: [1, 9, 7, 6, 5, 3] 	✓
<input type="radio"/>	<input checked="" type="radio"/>	<ul style="list-style-type: none"> • Ausgangsbelegung: [1, 3, 5, 8, 7, 2] • Mögliche Belegung: [1, 2, 5, 3, 8, 7] 	✓
<input checked="" type="radio"/>	<input type="radio"/>	<ul style="list-style-type: none"> • Ausgangsbelegung: [9, 7, 5, 4, 2, 1] • Mögliche Belegung: [1, 2, 5, 4, 7, 9] 	✓

- Ausgangsbelegung: [3, 6, 8, 2, 5, 4]
- Mögliche Belegung: [2, 3, 4, 6, 8, 5]
- : Falsch
- Ausgangsbelegung: [9, 7, 6, 5, 3, 1]
- Mögliche Belegung: [1, 9, 7, 6, 5, 3]
- : Falsch
- Ausgangsbelegung: [1, 3, 5, 8, 7, 2]
- Mögliche Belegung: [1, 2, 5, 3, 8, 7]
- : Falsch
- Ausgangsbelegung: [9, 7, 5, 4, 2, 1]
- Mögliche Belegung: [1, 2, 5, 4, 7, 9]
- : Richtig



Gegeben sei der aus den Vorlesungsfolien bekannte Sortieralgorithmus Insertionsort.

```
private static void insertionSort(int[] data) {
    for (int i = 1; i < data.length; i++) // Zeile A
    {
        for (int j = i; j > 0 && data[j] < data[j - 1]; j--) // Zeile B
        {
            exchange(data, j, j - 1); // Zeile C
        }
    }
}

private static void exchange(int[] a, int i, int j) { // Zeile D
    int swap = a[i];
    a[i] = a[j];
    a[j] = swap;
}
```

Beurteilen Sie für jede der unten angeführten Veränderungen, ob folgende Aussage richtig oder falsch ist: Der Algorithmus terminiert nach der Veränderung noch immer korrekt und sortiert alle möglichen `data`-Arrays mit einer Länge größer 0 noch immer aufsteigend.

Richtig Falsch

<input checked="" type="radio"/>	<input type="radio"/>	Zeile B: <code>for (int j = i; j > 0 && data[j - 1] > data[j]; j--)</code>	✓
<input checked="" type="radio"/>	<input type="radio"/>	Zeile A: <code>for (int i = 0; i < data.length; i++)</code>	✓
<input type="radio"/>	<input checked="" type="radio"/>	Zeile A: <code>for (int i = 0; i < data.length - 1; i++)</code>	✓
<input type="radio"/>	<input checked="" type="radio"/>	Zeile D: <code>int swap = a[j];</code>	✓

Zeile B: `for (int j = i; j > 0 && data[j - 1] > data[j]; j--)` : Richtig
 Zeile A: `for (int i = 0; i < data.length; i++)` : Richtig
 Zeile A: `for (int i = 0; i < data.length - 1; i++)` : Falsch
 Zeile D: `int swap = a[j];` : Falsch



Sie haben ein Array `data` mit n Werten gegeben.

Es soll ein Algorithmus verwendet werden, der überprüft, ob es zwei Werte gibt, deren Summe gleich k ist. Wenn ja, dann sollen die Indexpositionen dieser Werte in einem Array zurückgegeben werden. Ansonsten wird ein Array mit 2-mal -1 zurückgegeben.

Ihnen steht folgende Implementierung (Methode `check1`) zur Verfügung:

```
private static int[] check1(int[] data, int k) {
    for (int i = 0; i < data.length - 1; i++) {
        for (int j = i; j < data.length; j++) {
            if (data[i] + data[j] == k) {
                return new int[]{i, j};
            }
        }
    }
    return new int[]{-1, -1};
}
```

Daneben gibt es noch eine weitere Implementierung (Methode `check2`):

```
private static int[] check2(int[] data, int k) {
    int[] help = data.clone();
    Arrays.sort(help);
    int i = 0, j = help.length - 1;
    while (i < j) {
        int temp = help[i] + help[j];
        if (temp == k) {
            return new int[]{i, j};
        } else if (temp < k) {
            i++;
        } else {
            j--;
        }
    }
    return new int[]{-1, -1};
}
```

Bei der Methode `check2` wird eine Methode `sort` verwendet, die einen Sortieralgorithmus implementiert, dessen Laufzeit im Worst-Case in $O(n \log n)$ liegt.

Beurteilen Sie die Richtigkeit der folgenden Aussagen zu den beiden Methoden!

Richtig	Falsch		
<input checked="" type="radio"/>	<input type="radio"/>	Die Laufzeit von <code>check1</code> liegt im Worst-Case in $O(n^2)$.	✓
<input type="radio"/>	<input checked="" type="radio"/>	<code>check1</code> liefert nur eine korrekte Rückgabe, falls die Werte in <code>data</code> sortiert sind.	✓
<input type="radio"/>	<input checked="" type="radio"/>	<code>check2</code> liefert nur eine korrekte Rückgabe, falls <code>data</code> nur positive Werte beinhaltet.	✓
<input type="radio"/>	<input checked="" type="radio"/>	Beide Methoden verändern den Inhalt des Arrays <code>data</code> .	✓

Die Laufzeit von `check1` liegt im Worst-Case in $O(n^2)$.

: Richtig

`check1` liefert nur eine korrekte Rückgabe, falls die Werte in `data` sortiert sind.: Falsch

`check2` liefert nur eine korrekte Rückgabe, falls `data` nur positive Werte beinhaltet.: Falsch

Beide Methoden verändern den Inhalt des Arrays `data`.: Falsch

◀ 2. TUWEL-Test (13.01. B)

Direkt zu:

2. TUWEL-Test (20.01. B) ▶