

Algorithmen und Datenstrukturen

Übungsblatt 1

Rekursion, Arrays

Prof. Dr. Tim Reichert

1. Implementieren Sie eine Methode, welche alle natürlichen Zahlen bis zu einer gegebenen Zahl aufsummiert. Implementieren Sie eine **rekursive** Version und eine **iterative** Version mit Schleifen.

Beispiel:

```
sum(5)
Ergebnis: 15
```

```
sum(6)
Ergebnis: 21
```

2. Die folgenden Zahlen bilden den Anfang der Fibonacci-Folge:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

Diese Folge F_n ist definiert durch:

$$F_n = F_{n-1} + F_{n-2} \quad \text{für } n > 1 \quad \text{und} \quad F_0 = F_1 = 1$$

- 2.1 Implementieren Sie eine **rekursive** Methode zur Berechnung der Fibonacci-Folge. Wie bewerten Sie die Effizienz ihres Algorithmus?
- 2.2 Implementieren Sie entsprechend eine **iterative** Methode mit Schleifen. Wie bewerten Sie die Effizienz ihres Algorithmus?
3. Implementieren Sie den Euklidischen Algorithmus in der klassischen Version mit Subtraktion und in der modernen Version mit Modulo. Entwickeln Sie für beide Varianten eine **rekursive** Version und eine **iterative** Version mit Schleifen.
4. Implementieren Sie eine Methode, welche die Elemente eines Integer-Arrays summiert.

Beispiel:

```
int[] a = {1,2,3,4};
sumArray (a) ;
```

Ergebnis: 10

5. Implementieren Sie eine Methode, welche das größte Element eines Integer-Arrays zurückgibt.

Beispiel:

```
int[] a = {3,2,4,1};
max (a) ;
Ergebnis: 4
```

6. Implementieren Sie eine Methode, welche die Reihenfolge der Elemente eines Arrays umdreht.

Beispiel:

```
int [] a = {1, 2, 3, 4};  
reverse (a) ;
```

Ergebnis: {4, 3, 2, 1}

7. Implementieren Sie eine Methode, welche für jeder Zeile eines zweidimensionalen Arrays den Durchschnittswert berechnet.

Beispiel:

```
double[][] a = {{1,2},{10,20},{-5,5}};  
average (a) ;
```

Ergebnis: {1.5, 15, 0}

8. Implementieren Sie eine Methode, welche die Multiplikation von Matrizen mit Skalaren implementiert.

Beispiel:

```
double[][] a = {{1,2},{3,4},{5,6}};  
double b = 2;  
multiply (a,b) ;
```

Ergebnis: {{2,4},{6,8},{10,12}};

9. Implementieren Sie eine Methode, welche zwei Matrizen multipliziert. Beide Matrizen sind quadratisch (Zeilenzahl = Spaltenzahl) und gleich groß.

Beispiel:

```
double[] a = {{1,2}, {3,4}};  
double[] b = {{5,6}, {7,8}};  
multiply (a,b) ;
```

Ergebnis: {{19,22}, {43,50}};

10. Implementieren Sie eine Methode, welche innerhalb eines Arrays das Subarray findet, für das die Summe der enthaltenen Elemente am größten ist. Die Methode gibt diese maximale Summe zurück.

Beispiel:

```
double[] a = {-2, 1, -3, 4, -1, 2, 1, -5, 4};  
maxSum (a) ;
```

Ergebnis: 6 (für das Subarray {4, -1, 2, 1})