

Aufgabe 4

(a) Betrachten Sie das folgende Code-Beispiel (in Java-Notation):

```
4  int mystery(int n) {
5      int a = 0, b = 0;
6      int i = 0;
7      while (i < n) {
8          a = b + i;
9          b = a;
10         i = i + 1;
11     }
12     return a;
13 }
```

Bestimmen Sie die asymptotische worst-case Laufzeit des Code-Beispiels in \mathcal{O} -Notation bezüglich der Problemgröße n . Begründen Sie Ihre Antwort.

Die asymptotische worst-case Laufzeit des Code-Beispiels in \mathcal{O} -Notation ist $\mathcal{O}(n)$.

Die `while`-Schleife wird genau n mal ausgeführt. In der Schleife wird die Variable i in der Zeile `i = i + 1`; inkrementiert. i wird mit 0 initialisiert. Die `while`-Schleife endet wenn i gleich groß ist wie n .

(b) Betrachten Sie das folgende Code-Beispiel (in Java-Notation):

```
5  int mystery(int n) {
6      int r = 0;
7      while (n > 0) {
8          int y = n;
9          int x = n;
10         for (int i = 0; i < y; i++) {
11             for (int j = 0; j < i; j++) {
12                 r = r + 1;
13             }
14             r = r - 1;
15         }
16         n = n - 1;
17     }
18     return r;
19 }
```

Bestimmen Sie für das Code-Beispiel die asymptotische worst-case Laufzeit in \mathcal{O} -Notation bezüglich der Problemgröße n . Begründen Sie Ihre Antwort.

`while`: n -mal
1. `for`: $n, n-1, \dots, 2, 1$
2. `for`: $1, 2, \dots, n-1, n$
 $n \times n \times n = \mathcal{O}(n^3)$

(c) Bestimmen Sie eine asymptotische Lösung (in Θ -Schreibweise) für die folgende Rekursionsgleichung:

$$T(n) = T\left(\frac{n}{2}\right) + \frac{1}{2}n^2 + n$$