

Einzelprüfung „Theoretische Informatik / Algorithmen / Datenstrukturen (nicht vertieft)“

Einzelprüfungsnummer 46115 / 2016 / Herbst

## Thema 2 / Aufgabe 2

(Methoden „*matrixSumme()*“ und „*find()*“)

**Stichwörter:** Algorithmische Komplexität (O-Notation)

Geben Sie jeweils die kleinste, gerade noch passende Laufzeitkomplexität folgender Java-Methoden im O-Kalkül (Landau-Notation) in Abhängigkeit von  $n$  und ggf. der Länge der Arrays an.

(a)

```
int matrixSumme(int n, int[][] feld) {
    int sum = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            sum += feld[i][j];
        }
    }
    return sum;
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_46115/jahr\\_2016/herbst/Komplexitaet.java](https://github.com/bschlangaul/examen/examen_46115/jahr_2016/herbst/Komplexitaet.java)

Lösungsvorschlag

Die Laufzeit liegt in  $\mathcal{O}(n^2)$ .

Begründung (nicht verlangt): Die äußere Schleife wird  $n$ -mal durchlaufen. Die innere Schleife wird dann jeweils wieder  $n$ -mal durchlaufen. Die Größe des Arrays spielt hier übrigens keine Rolle, da die Schleifen ohnehin immer nur bis zum Wert  $n$  ausgeführt werden.

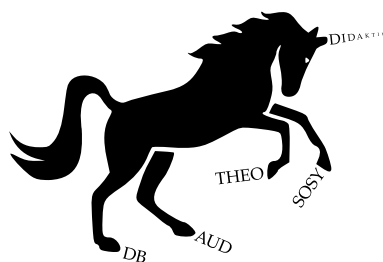
(b)

```
int find(int key, int[][] keys) {
    int a = 0, o = keys.length;
    while (o - a > 1) {
        int m = (a + o) / 2;
        if (keys[m][0] > key)
            o = m;
        else
            a = m;
    }
    return a;
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_46115/jahr\\_2016/herbst/Komplexitaet.java](https://github.com/bschlangaul/examen/examen_46115/jahr_2016/herbst/Komplexitaet.java)

Die Laufzeit liegt in  $\mathcal{O}(\log(\text{keys.length}))$ . Dabei ist `keys.length` die Größe des Arrays bezüglich seiner ersten Dimension.

Begründung (nicht verlangt): Der Grund für diese Laufzeit ist derselbe wie bei der binären Suche. Die Größe des Arrays bezüglich seiner zweiten Dimension spielt hier übrigens keine Rolle, da diese Dimension hier ja nur einen einzigen festen Wert annimmt.



## Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net). Der  $\text{\LaTeX}$ -Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Staatsexamen/46115/2016/09/Thema-2/Aufgabe-2.tex>