

**Name:**

**Matrikelnummer:**

**Bearbeitungszeit:** 60 min.

Schreiben Sie im Paket `filialnetz` die `public`-Klasse `Lagerauswahl` mit dem Konstruktor

```
public Lagerauswahl()
```

und der Methode

```
public int[] findeBesteLager(double[][] distanz).
```

Dabei gibt die Tabelle `distanz` die Distanzen zwischen einer Anzahl von Filialen an, wobei `distanz[i][j]` die Distanz zwischen Filiale `i` und `j` ist. Die Filialen sind mit  $0, \dots, n - 1$  nummeriert, die Anzahl der Filialen  $n$  ergibt sich als `distanz.length` und ist zumindest 2. Sie können annehmen, dass die Distanzen symmetrisch sind, also `distanz[i][j]==distanz[j][i]`.

Zwei der Filialen sollen zu Zwischenlagern ausgebaut werden. Dabei sollen die zwei Filialen so ausgewählt werden, dass die maximale Distanz von einer Filiale zu einem der beiden neuen Zwischenlagern möglichst klein ist.

Die Methode `findeBesteLager(distanz)` soll die Nummern der beiden optimalen Filialen in einem `int`-Array der Länge 2 zurückgeben. (Eventuell kann es auch mehrere gleich gute Lösungen geben. Eine davon soll zurück gegeben werden.)

*Hinweis:* Die besten zwei Filialen können durch ausprobieren gefunden werden.

*Beispiel:* Für die Distanzen

	0	1	2	3	4	5
0	0.0	5.0	6.0	8.0	3.0	8.0
1	5.0	0.0	8.0	7.0	8.0	9.0
2	6.0	8.0	0.0	4.0	5.0	8.0
3	8.0	7.0	4.0	0.0	7.0	6.0
4	3.0	8.0	5.0	7.0	0.0	7.0
5	8.0	9.0	8.0	6.0	7.0	0.0

ist  $\{0,3\}$  (bzw.  $\{3,0\}$ ) die einzige optimale Lösung. Die Distanz von jeder Filiale zu einem der Zwischenlager ist dann höchstens 6.0.