

## 186.866 Algorithmen und Datenstrukturen VU

# Programmieraufgabe P1

PDF erstellt am: 17. März 2023

## 1 Vorbereitung

Um diese Programmieraufgabe erfolgreich durchführen zu können, müssen folgende Schritte umgesetzt werden:

1. Laden Sie das Framework `P1.zip` aus TUWEL herunter.
2. Entpacken Sie `P1.zip` und öffnen Sie das entstehende Verzeichnis als Projekt in IntelliJ (nicht importieren, sondern öffnen).
3. Öffnen Sie die nachfolgend angeführte Datei im Projekt in IntelliJ. In dieser Datei sind sämtliche Programmieraktivitäten durchzuführen. Ändern Sie keine anderen Dateien im Framework und fügen Sie auch keine neuen hinzu.  
`src/main/java/exercise/StudentSolutionImplementation.java`
4. Füllen Sie Vorname, Nachname und Matrikelnummer in der Methode `StudentInformation provideStudentInformation()` aus.

## 2 Hinweise

Einige Hinweise, die Sie während der Umsetzung dieser Aufgabe beachten müssen:

- Lösen Sie die Aufgaben selbst und nutzen Sie keine Bibliotheken, die diese Aufgaben abnehmen.
- Sie dürfen beliebig viele Hilfsmethoden schreiben und benutzen. Beachten Sie aber, dass Sie nur die oben geöffnete Datei abgeben und diese Datei mit dem zur Verfügung gestellten Framework lauffähig sein muss.

## 3 Übersicht

In dieser Programmieraufgabe wird die Funktionsweise des bereitgestellten Frameworks sowie der allgemeine Ablauf von Programmieraufgaben in Algorithmen und Datenstrukturen nähergebracht.

Dazu werden einfache Algorithmen mit unterschiedlichen Laufzeiten aus der Vorlesung herangezogen.

## 4 Theorie

Die notwendige Theorie kann in den Vorlesungsfolien „Analyse von Algorithmen“ im Abschnitt „Laufzeiten einiger gebräuchlicher Funktionen“ gefunden werden.

## 5 Implementierung

### 5.1 Maximumsuche

Implementieren Sie die Maximumsuche in der Methode `int findMax(int[] numbers)`.

Mittels des Parameters `int[] numbers` wird ein Array an Zahlen übergeben, aus denen das Maximum ermittelt werden soll. Der Rückgabewert der Methode `int findMax(int[] numbers)` soll das gefundene Maximum sein. Gibt es kein Maximum, so geben Sie 0 zurück.

### 5.2 Dichtestes Punktepaar

Implementieren Sie Ihre Lösung zum dichtesten Punktepaar in der Methode `void findClosestPair(Point[] points, ClosestPair closestPair)`. Diese kann auf dem Pseudocode aus den Vorlesungsfolien basieren, die im Abschnitt 4 angegeben sind.

Im Parameter `Point[] points` wird ein Array mit mindestens zwei Punkten übergeben. Aus diesem soll ein Punktepaar mit minimaler euklidischer Distanz ermittelt werden.

Wird auf einen Punkt `Point point` aus dem Array zugegriffen, können mittels `int xPosition = point.getX()` bzw. `int yPosition = point.getY()` jeweils entsprechend die x-Position bzw. die y-Position ausgelesen werden.

Das gefundene Punktepaar soll im zweiten Parameter `ClosestPair closestPair` abgelegt werden. Dieser Parameter ist bereits initialisiert und stellt eine einfache Datenstruktur da, in der zwei `Points` abgelegt werden können. Angenommen die zwei ermittelten Punkte sind `Point pointA` und `Point pointB`, dann kann einer der beiden mit `closestPair.setPoint1(pointB);` und der andere mit `closestPair.setPoint2(pointA);` hinterlegt werden (und umgekehrt — die Reihenfolge ist nicht relevant).

Ein Rückgabewert wird nicht erwartet.

### 5.3 Teilsummenproblem

Implementieren Sie Ihre Lösung für das Teilsummenproblem in der Methode `boolean hasSubsetSum(int sum, int[] numbers)`.

Mittels des Parameters `int[] numbers` wird ein Array an Zahlen übergeben. Es soll festgestellt werden, ob es eine Untermenge dieser Zahlen gibt, deren Summe der Elemente dem Wert im Parameter `int sum` entspricht. Gibt es eine solche Untermenge, dann soll der Rückgabewert der Methode `true` sein, ansonsten `false`.

## 6 Testen

Führen Sie zunächst die `main`-Methode in der Datei `src/main/java/framework/Exercise.java` aus.

Anschließend wird Ihnen in der Konsole eine Auswahl an Testinstanzen angeboten, darunter befindet sich zumindest `abgabe.csv`:

```
Select an instance set or exit:
[1] abgabe.csv
[0] Exit
```

Durch die Eingabe der entsprechenden Ziffer kann entweder eine Testinstanz ausgewählt werden oder das Programm (mittels der Eingabe

von 0) verlassen werden. Wird eine Testinstanz gewählt, dann wird der von Ihnen implementierte Programmcode ausgeführt. Kommt es dabei zu einem Fehler, wird ein Hinweis in der Konsole ausgegeben.

Relevant für die Abgabe ist das Ausführen der Testinstanz `abgabe.csv`.

Die weiteren Testinstanzen `find-max-only.csv`, `find-closest-pair.csv` und `subset-sum-only.csv` sind nur zum jeweiligen Testen der einzelnen Unteraufgaben gedacht. Wenn Sie möchten, können Sie auch die Testinstanz `subset-sum-only-large.csv` ausführen, welche etwas Zeit in Anspruch nehmen wird.

## 7 Evaluierung

Wenn der von Ihnen implementierte Programmcode mit der Testinstanz `abgabe.csv` ohne Fehler ausgeführt werden kann, dann wird nach dem Beenden des Programms im Ordner `results` eine Ergebnis-Datei mit dem Namen `solution-abgabe.csv` erzeugt.

Die Datei `solution-abgabe.csv` beinhaltet Zeitmessungen der Ausführung der Testinstanz `abgabe.csv`, welche in einem Web-Browser visualisiert werden können. (Auch Ergebnis-Dateien anderer Testinstanzen können zu Testzwecken visualisiert werden.) Öffnen Sie dazu die Datei `visualization.html` in Ihrem Web-Browser und klicken Sie rechts oben auf den Knopf *Ergebnis-Datei auswählen*, um `solution-abgabe.csv` auszuwählen.

Beantworten Sie basierend auf der Visualisierung die Fragestellungen aus dem folgenden Abschnitt.

## 8 Fragestellungen

Öffnen Sie `solution-abgabe.csv` und bearbeiten Sie folgende Aufgaben- und Fragestellungen:

1. Durch Klicken auf Gruppennamen in der Legende neben der Plots, lassen sich einzelne Gruppen aus- bzw. einblenden. Blenden Sie alles bis auf *Maximumsuche* aus. Beschreiben Sie das Laufzeitverhalten anhand des Plots der *Maximumsuche* und erklären Sie basierend auf

Ihrer Implementierung, wieso diese zustande kommt. Stimmen die Messungen mit den Angaben aus der Vorlesung überein?

Drücken Sie im Anschluss in der Menüleiste rechts über dem Plot auf den Fotoapparat, um den Plot als Bild zu speichern.

2. Blenden Sie nun alles bis auf *Dichtestes Punktepaar* aus. Beschreiben Sie das Laufzeitverhalten anhand des Plots für *Dichtestes Punktepaar* und erklären Sie basierend auf Ihrer Implementierung, wieso diese zustande kommt. Stimmen die Messungen mit den Angaben aus der Vorlesung überein?

Erstellen Sie im Anschluss wieder mithilfe der Menüleiste ein Bild des Plots.

3. Blenden Sie nun alles bis auf *Teilsummen - Negativ* aus. Beschreiben Sie das Laufzeitverhalten anhand des Plots für *Teilsummen - Negativ* und erklären Sie basierend auf Ihrer Implementierung, wieso diese zustande kommt. Stimmen die Messungen mit den Angaben aus der Vorlesung überein?

Blenden Sie nun auch *Teilsummen - Positiv* und vergleichen Sie den Plot mit *Teilsummen - Negativ*. Sind größere Unterschiede zu sehen? Wenn ja, wie erklären Sie sich diese?

Erstellen Sie nun ein Bild der beiden Plots.

Falls sich im Zuge der Evaluierung die Darstellung der Plots auf ungewünschte Weise verändert (z.B. durch die Auswahl eines zu kleinen Ausschnitts), können Sie mittels Doppelklick auf den Plot oder Klick auf das Haus in der Menüleiste die Darstellung zurücksetzen.

Fügen Sie Ihre Antworten in einem Bericht gemeinsam mit den drei erstellten Bildern der Visualisierungen der Testinstanz `abgabe.csv` zusammen.

## 9 Abgabe

Laden Sie die Datei `src/main/java/exercise/StudentSolutionImplementation.java` in der TUWEL-Aktivität *Hochladen Source-Code P1* hoch. Fassen Sie diesen Bericht mit den anderen für das zugehörige Abgabegespräch relevanten Berichten in einem PDF zusammen und geben Sie dieses in der TUWEL-Aktivität *Hochladen Bericht Abgabegespräch 1* ab.