

Übungsblatt 1

Abgabe der Lösungen bis zum 24.11.2023 um 18:00 Uhr im Gitlab vom IBR. Es werden nur ausführbare und rechtzeitig eingereicht Lösungen bewertet. Python-Programme müssen mit python3 interpretierbar, und Java-Programme mit Java 11 kompilierbar sein.

Sofern nicht anders in der Aufgabe beschrieben, soll im Git die Ordnerstruktur wie vom Blatt vorgegeben beibehalten werden. D.h. für eine Aufgabe x von Blatt y sollen die verwendeten Source-Dateien (.py oder .java) in dem Ordner `blatt[y]/pflichtaufgabe[x]/` gespeichert werden.

Hausaufgabe 1 (Git Kenntnisse):

(3 Punkte)

In dieser Aufgabe wollen wir die grundlegenden Git-Befehle üben (Siehe auch Übung 1 vom 09.11.23). Dazu sollen beide Personen der Gruppe, nennen wir sie A und B, an ihrem Repository arbeiten.

Für den Fall, dass nur ein Rechner zur Verfügung steht, kann Person A die auf dem Rechner geklonte Version über Commandline, und Person B die Weboberfläche nutzen.

- a) Person A klonet das Git-Repository auf dem Rechner. Falls nicht bereits existent, legt dann eine `Readme.md` ohne Inhalt im root-Verzeichnis an.

Anschließend müssen diese Änderungen an den Server übermittelt werden. Zur Erinnerung:

- `git add file` bestimmt, dass eine Datei mit dem Namen *file* zum lokalen Repository hinzugefügt wird (oder Änderungen übermittelt werden).
- `git status` zeigt alle Dateien, welche Änderungen beinhalten oder nicht versioniert sind.
- `git commit -m "text"` erstellt den commit.
- `git push` übermittelt den commit an das Online-Git.

- b) Person B aktualisiert das Git-Repository auf dem anderen Rechner (im Webinterface einfach indem die Seite neu geladen wird) mit `git pull`.

Person B ändert dann die Datei `Readme.md`, sodass die erste Zeile den vollständigen Namen die Matrikelnummer, den Studiengang, sowie den geplanten Abschluss enthält.

Danach sollen die Änderungen wieder an den Server übermittelt werden (`git add`, `git commit` und `git push`).

- c) Person A ändert im für sich geklonten Repository auch die `Readme.md` mit gleichem Inhalt, **ohne vorher git pull** auszuführen.

Danach sollen die Änderungen an den Server übertragen werden. Es sollte nun die Nachricht erscheinen, dass ein Merge-Konflikt existiert, da online bereits eine andere Version von Person B vorliegt.

Löst den Konflikt und pusht die Datei erneut. Schaut euch nun in der Weboberfläche unter Code → Repository Graph an, was der Merge-Konflikt verursacht hat: Es wurde automatisch ein Branch erstellt, der über die Auflösung des Konflikts wieder auf den Master-Branch gemergt wurde.

Hausaufgabe 2 (Double Trouble (Python)): **(5 Punkte)**

In dieser Aufgabe soll ein Python-Programm entwickelt werden. Wie in der Vorlesung gelernt, können Vergleiche zwischen zwei double-Werten Schwierigkeiten bereiten. In dieser Aufgabe lernen wir ein zweites Problem kennen. Es sollen zwei Methoden (a und b) entwickelt werden, welche in der Datei `summe.py` implementiert werden sollen.

- a) Entwirf eine Methode `sumArray(array)`, welche alle Werte im übergebenen Array zusammenaddiert und die Summe zurückgibt.
- b) Entwirf eine Methode `sumArraySorted(array)`, welche alle Werte im übergebenen Array zusammenaddiert, wobei die Werte in aufsteigender Reihenfolge betrachtet werden sollen. Die Methode soll den Wert der Summe zurückgeben.

Um ein Array zu sortieren, kann die Anweisung `array.sort()` benutzt werden.

- c) Lass das Python-Programm in der Datei `main.py` ein Array erstellen, welches aus $10^7 + 1$ vielen Werten besteht. Dabei soll das Array zuerst einmal den Wert $2 \cdot 10^8$, und danach 10^7 mal den Wert 10^{-7} enthalten. Gib sowohl die Ergebnisse von `sumArray(array)`, `sumArraySorted(array)` und die Differenz von beiden auf der Konsole aus. Die jeweiligen Methoden sollen dabei nur ein einziges Mal aufgerufen werden. Gib außerdem die positive Differenz als Kommentar in der ersten Zeile von `summe.py` an (es reicht den Wert bis zur ersten Nachkommastelle anzugeben, welche nicht 0 ist).

Um ein Array mit n Feldern zu erzeugen, sodass alle Felder einen Wert x erhalten, kann zum Beispiel die Anweisung `arr = [x]*n` benutzt werden.

Anmerkung: Wie man sehen wird, kann die resultierende Abweichung sehr groß werden. Vor allem dann, wenn noch Subtraktion benutzt wird, wodurch der relative Fehler beliebig groß werden kann! Betrachte bspw. den Fehler, wenn am Ende noch der größte Wert aus dem Array subtrahiert wird.

Hausaufgabe 3 (Suchstrategien (Python)): **(7 Punkte)**

In dieser Aufgabe soll ein Python-Programm entwickelt werden. Wir wollen zwei Methoden vergleichen, die nach einer bestimmten Zahl in einem Array suchen. Diese zwei Methoden (a) und b)) sollen in der Datei `raten.py` implementiert werden.

- a) Implementiere eine Methode `iterativeSearch(array, num)`, welche ein Array `array` und eine Zahl `num` als Parameter erhält. Es soll nun überprüft werden, ob die Zahl `num` in `array` existiert, indem das Array iterativ durchsucht wird (also erst

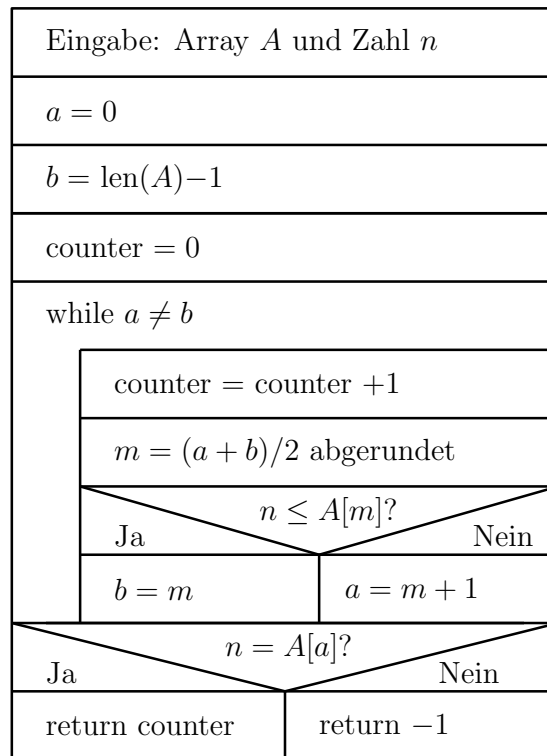


Abbildung 1: Ein Struktogramm zur binären Suche auf einem Array.

array[0], dann array[1], usw.). Wenn num in array existiert, soll die Methode am Ende die Anzahl an Iterationen zurückgeben. Existiert die gesuchte Zahl nicht, soll -1 zurückgegeben werden.

- b) Entwirf eine Methode `binarySearch(array, num)`, welche ein Array array und eine Zahl num als Parameter erhält. Auf dem Array soll eine binäre Suche durchgeführt, bis die gewünschte Zahl num gefunden wurde. Wenn num in array existiert, soll die Methode am Ende die Anzahl an Iterationen zurückgeben. Existiert die gesuchte Zahl nicht, soll -1 zurückgegeben werden.

Binäre Suche (siehe auch Abbildung 1) wird folgendermaßen durchgeführt: Betrachte einen Indexbereich a bis b für ein Array A . Ist die gesuchte Zahl kleiner oder gleich als $A[m]$ mit $m := (a + b) / 2$ (abgerundet), wird auf dem Bereich a bis m weitergesucht. Ist die Zahl größer als $A[m]$, wird im Bereich $m + 1$ bis b weitergesucht. Besteht der Wertebereich aus einer einzigen Zahl, ist dies die gesuchte Zahl, oder sie existiert nicht.

- c) Vervollständige das Programm in der `main.py`, in welchem ein Array mit 32 Werten erstellt wird, welche aufsteigend sortiert sind. Dann soll mit beiden Suchmethoden einmal nach einem Wert gesucht werden, der im Array existiert und einmal nach einem Wert, der nicht im Array existiert. Das Programm soll auf der Konsole ausgeben, wie viele Iterationen die beiden Suchstrategien benötigt haben bzw. dass die gesuchte Zahl nicht im Array existiert.

(Hinweis: Die Länge eines Arrays array erhält man über `len(array)`)