

Abgabe 2

Abgabe: 05.06.2022

Ziel der Aufgabe: In dieser praktischen Programmieraufgabe sind folgende Algorithmen und Datenstrukturen in der vorgegebene Java-Klasse zu implementieren, passend zu den entsprechenden Kapiteln der Vorlesung:

- a) **HASHMAP (MAX. 10 PUNKTE):** Implementieren Sie die Datenstruktur einer Hashtabelle, die Key-Value-Paare abspeichern kann, wobei der Schlüsselwert immer eine ganze Zahl ist, die mittels offenem Hashverfahren in ein Array eingefügt wird, und damit auch den Speicherplatz des entsprechenden Wertes bestimmt.

- Lineares Sondieren (max. 2 Punkte);
- Quadratisches Sondieren (max. 4 Punkte);
- Doppeltes Hashen (max. 4 Punkt).

Ihre Datenstruktur bekommt eine Mindestanzahl an Elementen übergeben, die darin Platz finden sollen. Dementsprechend ist eine Größe des zugrundeliegenden Arrays zu wählen (und mit `initTable` zu setzen), sodass die Sondierfunktionen gut arbeiten. Dies ist garantiert:

- wenn beim linearen Sondieren die entsprechende Mindestanzahl als Arraygröße verwendet wird,
- wenn beim quadratischen die nächstgrößere Primzahl als Arraygröße gewählt wird, die Kongruent 3 modulo 4 ist (also etwa 3, 7, 11, 19, 23, etc.), und
- wenn beim doppelten Hashen die nächstgrößere Primzahl als Arraygröße gewählt wird.

Als Hashfunktion ist die Funktion $h(n) = n \pmod k$ zu verwenden, wobei k die oben beschriebene Größe des zugrundeliegenden Arrays ist. Für das doppelte Hashen ist als zweite Hashfunktion die Funktion $h'(n) = (n \pmod k)' + 1$ zu verwenden, wobei k' die nächstkleinere Primzahl vor k ist.

- b) **QUICKSELECT (MAX. 5 PUNKTE):** Implementieren Sie den Quickselect-Algorithmus, wahlweise mit der entsprechenden Erweiterung zum Median-of-Medians-Algorithmus. Ihr Verfahren soll grundsätzlich In-Place (also mit max. $O(1)$ zusätzlichem Speicher) arbeiten; ausgenommen davon ist natürlich die Berechnung des Arrays der Mediane der Fünfer-Gruppen.

- Pivotelement am rechten Rand (max. 2 Punkte, entspricht "Quickselect");
- Pivotelement wird mittels Median-of-Medians bestimmt (max. 3 zusätzliche Punkte).

Vorbereitung: Laden Sie die in Moodle bereitgestellte Ab2-ZIP-Datei herunter. Es wird das Build-Tool `gradle` verwendet, welches Sie beim Arbeiten und auch beim Erzeugen Ihrer hochzuladenden ZIP-Datei unterstützt. Die ZIP-Datei enthält folgende Bestandteile:

- Interface **Ab2** im Java-Paket **ab2**.
- “Leere” Implementierung (als Grundgerüst) des oben genannten **Ab2**-Interfaces im Paket **ab2.impl.Nachnamen** (im Folgenden *Implementierungspaket* genannt). Ihre Aufgabe besteht in der Implementierung des entsprechenden Interfaces.
- Eine Testklasse **Ab2Tests** im Paket **ab2.test**, mit der Sie Ihre Implementierungen testen können. Diese Testklasse (mit zusätzlichen Testfällen) wird auch zur Bewertung Ihrer Abgabe verwendet (siehe Punkt “Testen”).

Integration in Ihre IDE (optional): Sie können (zB in Eclipse bzw. IntelliJ) den Inhalt der ZIP-Datei als Gradle-Projekt öffnen. Ihre IDE hat damit alle nötigen Informationen.

Durchführung: Die Programmieraufgabe kann in Gruppen bis zu 3 Personen bearbeitet werden. Alle Namen müssen in der Abgabe enthalten sein (siehe unten). Bei mehr als 3 Personen werden die erreichten Punkte anteilmäßig pro Person reduziert. Die geforderten Algorithmen sind im Implementierungspaket (impl) ohne Verwendung von Java-Bibliotheken und Systempaketen (außer **java.lang**) zu implementieren. Insbesondere darf das **java.util** Paket nicht verwendet werden. An den vorgegebenen Interfaces und an der Testklasse dürfen, bis auf die Umbenennung Ihres Implementierungspakets, keine Änderungen vorgenommen werden. Ändern Sie ausschließlich die Datei “Ab2Impl.java” (bzw., wenn benötigt, können auch neue Dateien im Ordner “ab2/impl” erstellt werden). Bei Unklarheiten bezüglich der Aufgabenstellung kontaktieren Sie bitte Ihren LV-Leiter.

Testen: Rufen Sie **./gradlew test** bzw. **gradlew.bat test** auf um die Testfälle durchlaufen zu lassen. Alternativ können Sie die Tests auch in Ihrer Entwicklungsumgebung laufen lassen. Zum Bewerten Ihrer Abgabe werden zusätzliche, nicht ausgegebene Testfälle verwendet. Es ist daher von Vorteil, wenn Sie selbst zusätzliche Testfälle implementieren, und Ihre Lösung damit testen.

Abgabe: Rufen sie **./gradlew zip** bzw. **gradlew.bat zip** auf. Sie finden nun die hochladbare ZIP-Datei im Ordner **build**. Geben Sie die ZIP-Datei via Moodle ab. Pro Gruppe ist nur eine Abgabe nötig. Benennen Sie vor der Abgabe das Paket **ab2.impl.Nachnamen** entsprechend um, indem Sie **Nachnamen** durch die Nachnamen Ihrer Gruppenmitglieder ersetzen (z.B. **ab2.impl.HuberMeierMueller**).

Bewertung: Diese Programmieraufgabe wird mit maximal 15 Punkten bewertet. Die erreichte Punktezahl wird allen Gruppenmitgliedern gleichermaßen angerechnet. Die Bewertung ergibt sich aus folgenden Komponenten:

- Funktionalität Ihres Codes. Die Anzahl erfolgreicher Unit-Tests sind hierfür ausschlaggebend. Beachten Sie, dass für die Bewertung nicht nur die Testfälle der ausgegebenen Testklasse verwendet werden, sondern auch zusätzliche, erweiterte Testfälle. Beim Testen wird von Ihrer ZIP-Datei nur das Implementierungspaket verwendet.
- Qualität Ihres Quellcodes (Inspektion). Achten Sie auf kommentierten, gut verständlichen und effizienten Quellcode. Die Verwendung von **System.out** ist nicht gestattet.
- Ehrlichkeit: Falls Sie unzulässige Java-Bibliotheken (siehe oben) verwenden, oder Sie den selben Quellcode wie eine andere Gruppe abgeben, kann Ihre Abgabe mit 0 Punkten bewertet werden.
- Formatierung: Formatieren Sie Ihren Code (Standard-Formatierung Ihrer IDE ist ausreichend) und achten Sie darauf, dass sie keine nicht verwendeten Imports haben.