

Abgabe 1

In dieser Abgabe beschäftigen wir uns mit Heapsort. Bearbeiten Sie dazu die folgenden Teilaufgaben bis zum **Mittwoch 8.6.2022 um 23:59 CEST**.

Aufgabe 1 (Heapsort)

Implementieren Sie eine Heapsortvariante, die Strings aufsteigend sortiert. Entwickeln Sie einen einfachen Unit-Test, der durch eine geeignete Ausgabe belegt, dass Ihr Heapsort korrekt funktioniert. Beachten Sie, dass Lösungen, die entgegen der Aufgabenstellung, nur ganze Zahlen sortieren zu einer nicht ausreichenden Wertung führen!

Aufgabe 2 (Zufällige Strings)

Entwickeln Sie eine Methode/Funktion `String getRandomString(int len)` die zufällige Strings einer gegebenen Länge liefert.

Aufgabe 3 (Benchmark)

Führen Sie Laufzeittests mit Ihrer Heapsort-Implementierung aus. Bestimmen Sie dazu die Sortierzeit für $n \in \{100000, 200000, 300000, 400000, 500000, 600000, 700000, 800000, 900000, 1000000, 2000000, 4000000, 8000000\}$ Strings der Länge von genau 16. Evtl. müssen Sie das Sortieren für ein n mehrfach wiederholen, wenn ein einzelner Durchlauf zu schnell fertig ist (Ein 3Ghz Intel Xeon sortiert mit einer unoptimierten Heapsortvariante 100000 `int`-Werte in 15ms). Erstellen Sie für die bestimmten Laufzeiten geeignetes Diagramm. Welche Komplexität vermuten Sie? Bestimmen Sie anhand der Laufzeiten die Konstante c überschlagsweise, die durch die O-Notation verborgen wird.

Eine Zeitmessung könnte (grob!) wie folgt aussehen:

```
/* get start time */
(void) gettimeofday(&before, &zone);

/* Sort the data */
HeapSort(A, length);

/* get end time */
(void) gettimeofday(&after, &zone);

/* Add the time needed for the heap-sort call (converted to ms) */
elapsedTime += 1000 * (after.tv_sec - before.tv_sec) +
((after.tv_usec - before.tv_usec) / 1000);
```

bzw.

```
long start = System.nanoTime();

HeapSort(A)

long end = System.nanoTime();
long microseconds = (end - start) / 1000;
```

Durchführung:

1. Bilden Sie maximal 2er Teams, wobei 3 oder mehr Personen in einem Team *nicht* zulässig sind.
2. Lösen Sie die Aufgaben 1-3.
3. Schreiben Sie ein README-File, das erklärt wie Ihr Projekt, ohne Hilfe einer IDE zu bauen ist. Kann Ihr abgegebenes Projekt nicht gebaut werden oder fehlen entsprechende Hinweise, so kann dies zu einer nicht ausreichenden Wertung führen! Packen Sie den Quellcode, das README und die Ergebnisse der Laufzeittests in ein .zip oder .tgz File.
4. Geben Sie Ihre Lösung (.zip/.tgz) diesmal via EMail bei Ihrem Gruppenleiter ab. Die Betreffzeile Ihrer EMail *muss* die folgende Form haben **Abgabe 1 ADS: Nachname1_Vorname1, Nachname2_Vorname2**. E-mails, die man nicht zuordnen kann, werden nicht gewertet.

Hinweis: Beachten Sie die Bemerkungen zum Thema *Plagiate*. Im Plagiatsfall bekommen mindestens *beide* beteiligte Parteien keine Punkte. Es könnten auch härtere Strafen in Betracht kommen.