

Aufgabe 1 – Quicksort

Vorbereitung

- Machen Sie sich anhand der Unterlagen aus der Vorlesung mit dem Konzept der Rekursion vertraut.
- Sehen Sie sich noch einmal das höhere Sortierverfahren Quicksort an.
- Untersuchen Sie die Lomuto-Partition des Quicksort.
- Überlegen Sie sich, wie man die Median-von-3 Variante der Lomuto Partition effizient implementieren könnte.

Aufgabenbeschreibung

1. Schreiben Sie ein Java-Programm mit dem Pseudozufallszahlen zeilenweise in eine Datei geschrieben werden. Diese Daten werden anschließend als Eingangsdaten verwendet. Erstellen Sie 3 Dateien mit 100.000, 1.000.000 und 5.000.000 Zufallszahlen mit Werten im Intervall zwischen 0 und 5.000.000.

Verwenden Sie zur Erzeugung der Pseudozufallszahlen die Klasse `Java.util.Random`

(siehe <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>).

2. Schreiben Sie ein Java-Programm zur Sortierung der Testdaten. Gehen Sie dabei vor:
 - Die Testdaten werden in ein Array eingelesen (Basisdatentyp `int`).
 - Dieses Array wird mit dem rekursiven Quicksort sortiert. Dabei wird die Standard-Lomuto Partition verwendet.
 - Die Dauer der Sortierung soll durch Aufruf der Methode `System.currentTimeMillis()` (siehe [https://docs.oracle.com/javase/7/docs/api/java/lang/System.html#currentTimeMillis\(\)](https://docs.oracle.com/javase/7/docs/api/java/lang/System.html#currentTimeMillis())) direkt vor und hinter der Sortiermethode ermittelt werden. Die Dauer ergibt sich als Differenz der beiden so bestimmten Zeitpunkte.
3. Ermitteln Sie die Dauer der Sortierung der drei Testdatensätze.
4. Ermitteln Sie die Dauer der Sortierung für 100.000 **bereits sortierter Daten**.
5. Schreiben Sie Ihr Programm so um, dass die Median-von-3 Variante der Lomuto Partition implementiert wird.
6. Ermitteln Sie die Zeiten wie unter 3. und 4..

Hinweise

Alle Laufzeiten müssen außerhalb des **Debug Modes** ermittelt werden. Lassen Sie Ihre Projekt daher nicht im Debug Modus laufen, da dies sehr viel mehr Rechenzeit benötigt.

Wegen einer Vielzahl von rekursiven Aufrufen könnte es eine Stackoverflow-Exception geben. Dann muss für diese Java-Anwendung die Größe des Stacks erweitert werden. Dies geschieht in den Entwicklungsumgebungen wie folgt:

Eclipse: Gehen Sie über **Run→Run Configurations→Java Application** zu den Einstellungen Ihres Projektes. Im Reiter **(x)= Arguments** tragen Sie in der Textbox **VM Arguments** den Text **-Xss10M** ein. Dies erweitert den Stack auf 10Mbyte.

IntelliJ: Gehen Sie über **Run→Edit Configurations** zu den Einstellungen Ihres Projektes. Tragen Sie im Reiter **VM Options:** wieder **-Xss10M** ein.

Netbeans 11: Durch einen Rechtsklick auf dem Projekt klappt ein Menü aus. Dann dort **Properties** auswählen und dort bei der Kategorie **Run** unter **VM Options:** wieder **-Xss10M** eingeben.

Abgabe

Laden Sie das Programm ihrer Gruppe im Dateiordner ‚Abgabe A1‘ im Dateibereich von Stud.IP hoch. Bilden Sie dazu ein ZIP-Archiv, dass die Quelldateien und eine Textdatei mit der Laufzeiten enthält. Der Name des Archivs muss die Namen der Gruppenmitglieder enthalten, also z.B. **Meyer_Schulze.zip**. Dieses Archiv muss zu Beginn Ihres nächsten Praktikumstermins in OSCA vorliegen.

Bitte beachten Sie die allgemeinen Hinweise zum Praktikum.