EIGENE MESSUNGEN ZU AUFGABE 2

TEAM 10: Anton, IGOR & mesut

# EINLEITUNG

Die Aufgabe 2 ,,naives vs komplexes Sortieren“ bei der Veranstaltung ,,Algorithmen und Datenstrukturen“ (HAW Hamburg) erforderte, dass wir eine endliche Folge von positiven Zahlen in einer erzeugten Datei ,,zahlen.dat“ *(oder vorgegebene ,,zahlen.dat“)* nach bestimmten Sortieralgorithmen implementieren – basierend auf unsere vorherige Implementation von ADTArray. Zu den Sortieralgorithmen haben wir ebenfalls Laufzeit und Zugriffsmessungen erstellt, sodass wir beurteilen können, welches Sortierverfahren wann bestens geeignet ist.

## messungen

Um möglichst aussagekräftige Messungen durchzuführen, haben wir bei der Implementation darauf geachtet, dass die erstellte .csv-Datei *(interne Vorgabe)* auch Übersichtlich gestaltet ist, sodass dementsprechend ein Graph bei Excel nach Vorgabe erstellt werden kann. Die unten gezeigten Messungen in der Tabelle sind nur herausgenommene Beispieldaten aus der ,,benchmark.csv“ -Datei.

Bei den Resultaten auf folgendes aufpassen:

* Angenommen: die Laufzeit bei Quicksort ,,**200 Zahlen,63,16,79**“ *(die Messung befindet sich bei der Tabelle an der vierten Stelle)*. Die 63 ist dabei die Zeit, die Quicksort selbst braucht, die 16 für InsertionSort und die 79 ist die gesamte Zeit für den Aufwand.

|  |  |  |  |
| --- | --- | --- | --- |
| Versuchsaufbau | Algorithmus | Resultate | Kommentar |
| Laufzeit | Insertion Sort (random number) | ../../../../../../../Desktop/Bildschirmfoto%202015-11-18%20um% | Insertion Sort braucht für zufällige Zahlen bedauerlicherweise sehr viel. |
| Laufzeit | Insertion Sort (left sorted) | ../../../../../../../Desktop/Bildschirmfoto%202015-11-18%20um% | Für linkssortierte Zahlen braucht Insertion Sort sich nicht mal anstrengen. |
| Laufzeit | Insertion Sort (right sorted) | ../../../../../../../Desktop/Bildschirmfoto%202015-11-18%20um% | Bei rechtssortierten Zahlen haben wir noch mal bei größeren Zahlen einen aufwändigeren Insertion Sort. |
| Laufzeit | Quicksort (pivot links – random numbers) | ../../../../../../../Desktop/Bildschirmfoto%202015-11-18%20um% | Quicksort braucht bei einem pivot links mit zufälligen Zahlen bei beispielsweise 200 Zahlen nur 63 ms. |
| Zugriffe | Insertion Sort *(random number)* | ../../../../../../../Desktop/Bildschirmfoto%202015-11-19%20um% | Zugriffe bei Insertion Sort sind enorm viel, gerade bei zufälligen Zahlen ist das allerdings vorhersehbar. |
| Zugriffe | Quick Sort (pivot links) | ../../../../../../../Desktop/Bildschirmfoto%202015-11-19%20um% | Quick Sort mit einem pivot links braucht großen Aufwand (bei 200 Zahlen 5934ms) ohne Insertion Sort. |
| Zugriffe | Quick Sort (pivot random) | ../../../../../../../Desktop/Bildschirmfoto%202015-11-19%20um% | Quick Sort mit einem zufälligen Pivot braucht allerdings auch einen großen Aufwand. |