Statistische Auswertung für die Programmieraufgabe 3

In diesem Projekt wurden vier verschiedene Suchalgorithmen in Java implementiert und getestet:

- Lineare Suche *(linearSearch*)
- Binäre Suche (*binarySearch*)
- Interpolationssuche (*interpolationSearch*)
- Quadratische binäre Suche (*quadraticBinarySearch*)

Ziel war es, die Effizienz der Algorithmen auf sortierten Integer-Arrays unterschiedlicher Größe zu vergleichen.

Testmethode

Für jeden Algorithmus wurden alle Werte von 1 bis n in Arrays der Größen gesucht:

- 100
- 1.000
- 10.000
- 100.000

Dabei wurden folgende Kennzahlen gemessen:

- Minimale Laufzeit (ns)
- Maximale Laufzeit (ns)
- Durchschnittliche Laufzeit (ns)

Das Ergebnis wurde in der Datei evaluation/statistik.txt gespeichert.

Ergebnisse (Auszug)

===== Array Size: 100 =====

Linear Search -> Min: 200 ns | Max: 868100 ns | Average: 10226,00 ns

Binary Search -> Min: 200 ns | Max: 8300 ns | Average: 516,00 ns

Interpolation Search -> Min: 200 ns | Max: 12200 ns | Average: 402,00 ns

QuadraticBinary Search -> Min: 200 ns | Max: 8800 ns | Average: 610,00 ns

===== Array Size: 1000 =====

Linear Search -> Min: 200 ns | Max: 495000 ns | Average: 2647,10 ns

Binary Search -> Min: 100 ns | Max: 247700 ns | Average: 709,10 ns

Interpolation Search -> Min: 100 ns | Max: 830500 ns | Average: 1127,40 ns

QuadraticBinary Search -> Min: 100 ns | Max: 798300 ns | Average: 1270,10 ns

===== Array Size: 10000 =====

Linear Search -> Min: 200 ns | Max: 32167000 ns | Average: 6022,11 ns

Binary Search -> Min: 100 ns | Max: 3185600 ns | Average: 622,81 ns

Interpolation Search -> Min: 100 ns | Max: 231000 ns | Average: 176,31 ns

QuadraticBinary Search -> Min: 100 ns | Max: 44400 ns | Average: 275,39 ns

Fazit

Die lineare Suche hat deutlich höhere Laufzeiten bei großen Arrays.

Die binäre Suche ist effizient und konstant schnell.

Interpolationssuche war besonders schnell bei gleichmäßig verteilten Daten.

Quadratische binäre Suche ist ebenfalls effizient, jedoch leicht langsamer als Interpolation.