Bern, 28.03 Winkler Olivier & Winkelmann Domenico

Sortier-algorithmen

# Einleitung

Erklären Sie hier in eigenen Worten und maximal zwei Sätzen, was ein Sortieralgorithmus tut.

# Selection Sort

Beschreiben Sie hier, wie der Selection Sort Algorithmus arbeitet. Beschreiben Sie in ganzen Sätzen.

|  |  |
| --- | --- |
| Sie können sich an folgendem englischen Text nebenan orientieren. |  |

Zur Veranschaulichung betrachten wir folgendes Array:

int[] a = {11, 9, 17, 5, 12}; sortiert unsortiert

Vervollständigen Sie den Durchlauf mit Ihren Kommentaren.

|  |  |
| --- | --- |
| 11 9 17 5 12 | Initial Array, kleinstes Element 5, kommt an die Position a[0], indem wir a[0] mit a[3] vertauschen |
| 5 9 17 11 12 | Im unsortierten Bereich ist die 9 die kleinste Zahl, welche bereits an richtiger Stelle steht. |
| 5 9 17 11 12 | Ihre Beschreibung  ................................................. |
| 5 9 11 17 12 | Ihre Beschreibung  ................................................. |
| 5 9 11 12 17 | Der unsortierte Bereich besteht aus noch einem einzigen Element, also sind wir fertig. |
| 5 9 11 12 17 |  |

# Implementation des selection sort Algorithmus

Implementieren Sie den Selection Sort Algorithmus in einer eigenen Klasse und kopieren Sie den Source Code hier hin. Tun Sie das, indem Sie eine Klasse SelectionSorter erstellen nach folgenden Vorgaben:

public class SelectionSorter {

public SelectionSorter(int[] myArray) {

a = myArray;

}

public void sort() {

<hier kommt Ihre Implementation>

}

private int[] a;

}

# Selection Sort Testprogramm

Damit wir unsere Klasse SelectionSorter testen können, implementieren Sie folgende Klasse:

public class SelectionSortTester {

public static void main(String[] args) {

// Erstellen Sie hier ein Array aus Zufallszahlen

int[] a = ....

// geben Sie das Array auf die Konsole aus.

// Verwenden Sie nun hier Ihren SelectionSorter

SelectionSorter sSorter = new SelectionSorter(a);

sSorter.sort();

// geben Sie das Array zur Kontrolle auf die Konsole aus.

}

}

# Profiling des Selection Sort Algorithmus

Analysieren Sie nun wie viel Zeit der Selection Sort Algorithmus beansprucht, um Arrays mit verschiedener Grösse zu sortieren. Ergänzen Sie dazu den SelectionSortTester mit einer Zeitmessung folgendermassen:

int[] a = ... // ein Array aus Zufallszahlen mit 10'000 Elementen.

SelectionSorter sSorter = new SelectionSorter(a);

long start = System.currentTimeMillis();

sSorter.sort();

long stop = System.currentTimeMillis();

und erstellen Sie eine Grafik gemäss untenstehender Vorlage.

|  |  |
| --- | --- |
|  | x-Achse: #Elemente im array  y-Achse: Zeit in ms zum Sortieren |
|  |  |

Fragen:

1. Versuchen Sie anhand der Grafik abzuschätzen wie lange das Sortieren von 120’000 Elementen dauert.
2. Welcher mathematischen Gesetzgebung könnte diese Funktion folgen?

# Vergleich mit Java Sort

Wiederholen Sie die Messungen mit der Java Funktion

Arrays.sort(int[] ints) aus dem Paket java.utils.Arrays.

und ergänzen sie die Resultate in Ihrer Grafik.

Frage:

1. Was stellen Sie fest? Wie erklären Sie sich das Ergebnis?