

Sortier-Algorithmen (K2, S2, S4)

Fällig 12. Jan 2024 bis 23:59 Punkte 0

Sortier-Algorithmen (K2, S2, S4)

Erstelle dir vor Beginn der Arbeit einen Zeitplan in Form einer Tabelle mit 3 Spalten. Auf diesem schreibst du dir die einzelnen Meilensteine deiner geplanten Umsetzung in die erste Spalte auf. In die zweite Spalte schreibst du auf wie lange deiner Meinung nach du für das abschließen des Meilensteins voraussichtlich brauchst. Bei deiner Arbeit misst du deine Zeit. Wenn der Meilenstein abgeschlossen ist, schreibst du in die dritte Spalte, wie lange du tatsächlich dafür gebraucht hast.

Erstelle ein Projekt, in welchem du eine Reihe an Zahlen auf verschiedene Weisen sortieren lässt. Erstelle zusätzlich passende Ein- und Ausgaben für die Verwendung zur Laufzeit.

Zu Beginn soll es die Möglichkeit eine von bestimmter Menge an zufälligen Zahlen generieren zu lassen bzw. selber eine Reihe an Zahlen einzugeben.

Diese Zahlen sollen dann mit mindestens 3 verschiedene Arten sortiert werden. Schau dir dafür unterschiedliche Algorithmen an und überlege dir, welche dafür geeignet sind. Die Auswahl soll zur Laufzeit gemacht werden können.

Die sortierte Reihe soll aufsteigend, absteigend oder im zickzack (zuerst die größte Zahl, dann die kleinste Zahl, dann die zweit größte und dann die zweit kleinste, usw.) ausgegeben werden. Die Auswahl soll ebenfalls zur Laufzeit gemacht werden können.

Tipps:

- Wende dich bei Problemen an dein Fachbereichs-Team
- Lass andere aus deinem Kurs deinen Code testen um mögliche Fehler zu finden.
- Bubblesort, Selectionsort, Quicksort, Insertionsort, Mergesort sind interessante Algorithmen für diese Aufgabe
- Vermeide den Bogosort!

Lernziele:

K2 - Erläutern von Algorithmen und deren Eigenschaften

S2 - Entwurf von Algorithmen für bestimmte Aufgaben

S4 - Planung von Terminen und Ressourcen, um Projektziele und -ergebnisse erfolgreich zu erreichen

Feedbackelemente:

Achte auf folgende Elemente und notiere dir dazu Feedback von deiner Fachbetreuung:

Proficiency:

- Läuft das Programm stabil und fehlerfrei?
- Ist die User-Eingabe verständlich gestaltet?
- Wurden für die geforderten Arten, die Zahlen zu sortieren, die am besten geeigneten Algorithmen verwendet?
- Sind die Algorithmen richtig implementiert?

Process:

- Wurden eine einheitliche Coding-Convention verwendet?
- Werden die Eingaben des Nutzers richtig angewendet?
- Wurden alle Methoden sinnvoll benannt?

Person:

- Wie gut ist der erstellte Code verständlich und lesbar?
- Wurden über den Unterricht hinausgehende Quellen verwendet?