Informatik: Sortieralgorithmen, Aufwandschätzungen

Ziele

- Folgende Sortierverfahren verstehen, programmieren und in ihren Aufwänden abschätzen können:
  - Sortieren durch *Minimumsuche*
  - Sortieren durch *Einordnen*
  - Sortieren durch *Mischen*
  - Quicksort *kombiniert mit anderen Verfahren*
  - Quicksort mit *geschickter Trennelementewahl*
  - *Bsort*
  - *Heapsort*
- Den günstigsten Sortieralgorithmus in Abhängigkeit des gestellten Problems auswählen können
- Eine Sortiererklasse programmieren können
- Im Team gemeinsam seine Arbeitsschritte organisieren und abstimmen können

Gegenüberstellung der Sortierverfahren

Verfahren	Mindestens	Im Mittel	Maximal
Sortieren durch Minimum-Suchen		$1 + 2 + \ldots + n = \frac{N^2}{2}$	
Sortieren durch Einordnen	N	$\frac{N^2}{4}$	$\frac{N^2}{2}$
Sortieren durch Mischen		$N \ ld \ N$	
Quicksort	$N \cdot ld \ N$	$1.4 \cdot N \cdot ld \ N$	$\frac{N^2}{2}$
Heapsort	im ungünstigsten Fall $2 \cdot N \cdot ld \ N$		

Teamarbeit Sortieralgorithmen

- In *Zweierteam* die zwei zugewiesenen Sortieralgorithmen exakt nach Vorgabe als Klasse ausprogrammieren
- Beide Sortieralgorithmen mit unterschiedlichen Datenmengen (sortiert, nicht sortiert) auf demselben Computer laufen lassen und Geschwindigkeit vergleichen
- Es müssen Methoden zum Füllen der Arrays bereitgestellt werden
- Sortieralgorithmen der *Partnergruppe* mit denselben Datenmengen auf demselben Computer laufen lassen und Geschwindigkeit vergleichen
- *Schnittstellendefinition* abgestimmt mit der Partnergruppe in schriftlicher Form innerhalb kürzester Zeit liefern
- Gemeinsames *Testprogramm* mit Partnergruppe
- Einsatz der Klasse `Stoppuhr` welche in einem der letzten Kapitel ausprogrammiert wurde
- Messungen sollen *logarithmisch* erfolgen
  (10, 20, …, 90, 100, 200, …, 900, 1.000, 2.000, …, 9.000,…)
- Sortieralgorithmus muss auch mit *größeren Datenmengen* getestet werden
- Es können `int`-Werte sortiert werden
- *Testbericht* der die Sortieralgorithmen gegenüberstellt in Form eines *Excel-Diagramms*