

Aufgabe 5d: Algorithmen und Datenstrukturen

1 Lernziele

Anwenden der modularen Programmierung, Vertiefen der Kenntnisse über die Gültigkeit und Sichtbarkeit von Variablen sowie Erfahrungen sammeln im gemeinsamen Erstellen von Software.

2 Allgemeines

Die Aufteilung, wer aus der Zweiergruppe welches Modul implementiert, bleibt Ihnen überlassen. Am Ende des Praktikums müssen alle Module zusammen kompilierbar sein und das Programm die gewünschte Funktionalität aufweisen.

Um Daten innerhalb eines großen Datenbestands schnell finden zu können, muss der Datenbestand nach bestimmten Kriterien geordnet sein. So ist es in einem Telefonbuch zwar möglich, sehr schnell zu einem Teilnehmer die Rufnummer zu ermitteln. Um aber zu einer gegebenen Rufnummer den Teilnehmer zu finden, muss evtl. das gesamte Telefonbuch durchsucht werden.

3 Aufgabe

Schreiben Sie ein Programm, das zwei unterschiedliche Sortierverfahren miteinander vergleicht. Dazu werden in einer Messreihe unterschiedlich viele Daten sortiert und eine Statistik erstellt. Die Statistik enthält die Anzahl der durchgeführten Vergleichs- und Vertausche-Operationen.

Teil 1: Implementieren Sie ein Modul `quicksort`, das beliebige Daten mittels des in der Vorlesung besprochenen Quicksort-Algorithmus sortiert. Verwenden Sie nicht die Funktion `qsort` aus der Standardbibliothek!

Teil 2: Implementieren Sie ein Modul `bubblesort`, das beliebige Daten mittels des unten stehenden Bubblesort-Algorithmus sortiert.

Bei jedem Sortiervorgang soll die Anzahl der durchgeführten Vergleichs- und Vertausche-Operationen gezählt werden. In beiden Fällen ist folgende Schnittstelle zu verwenden:

```
void sort(void *data, int nmemb, int size,
          int (*cmp)(void *, void *),
          int *moves, int *comps);
```

Das Hauptprogramm bekommen Sie von uns gestellt. In dem Hauptprogramm werden zufällige Daten erzeugt, der Sortierfunktion zum Sortieren übergeben und die Statistik erstellt und angezeigt.

4 Bubblesort

solange noch nicht alle Zahlen sortiert sind:

- durchlaufe die Zahlenfolge von links nach rechts
- vertausche ggf. zwei nebeneinander stehende Zahlen

Beispiel:

erster Durchlauf

44	55	12	42	94	18	06	67
44	55	12	42	94	18	06	67
44	12	55	42	94	18	06	67
44	12	42	55	94	18	06	67
44	12	42	55	94	18	06	67
44	12	42	55	18	94	06	67
44	12	42	55	18	06	94	67
44	12	42	55	18	06	67	94

zweiter Durchlauf

44	12	42	55	18	06	67	94
12	44	42	55	18	06	67	94
12	42	44	55	18	06	67	94
12	42	44	55	18	06	67	94
12	42	44	18	55	06	67	94
12	42	44	18	06	55	67	94
12	42	44	18	06	55	67	94

Die Paare, die miteinander verglichen und ggf. getauscht werden, sind hervorgehoben.

5 Testat

Voraussetzung ist jeweils ein fehlerfreies, korrekt formatiertes Programm. Der korrekte Programmlauf muss anhand einer Beispieleingabe nachgewiesen werden. Sie müssen in der Lage sein, Ihr Programm zu erklären.