
2. Übungsblatt

4. Aufgabe (Praktikum):

Implementieren Sie REKURSIVE-Varianten der Bubble-Sort und Quicksort-Algorithmen. Führen Sie dabei Zeitmessungen durch und zählen Sie, wie oft die unterschiedlichen Anweisungen durchgeführt werden.

Achten Sie neben der Korrektheit des Programms insbesondere auf die Effizienz (Geschwindigkeit, Vermeidung von überflüssigen Anweisungen) des Programms.

Zusätzlich werden folgende Informationen gesucht:

- a) Messen Sie wie lange es dauert,
 - die zu sortierenden Zahlen einzulesen,
 - die Zahlen zu sortieren und
 - die Zahlen sortiert in einer Datei zu schreiben sowie
 - das gesamte Programm auszuführen.
- b) Damit die Zeitmessung nicht verfälscht wird, zählen Sie in einem separaten Lauf
 - die Zuweisungen,
 - die Bedingungen und
 - Unterprogrammaufrufe (das Vertauschen 2er Zahlen soll in einem separaten Unterprogramm erfolgen),die während des Sortiervorgangs benötigt werden.

Zur Zeitmessung können Sie die folgende Prozedur verwenden, die in C zur Verfügung gestellt wird:

```
#include <dos.h>
void gettime(struct time *timep);
void settime(struct time *timep);
```

Beschreibung

Lesen und Setzen der Systemzeit.
gettime liefert in der über timep angegebenen
Struktur des Typs time die aktuelle Uhrzeit des

Systems zurück.

settime setzt die Uhrzeit des Systems auf die Werte, die in der über timep angegebenen Struktur des Typs time enthalten sind.

Der Strukturtyp time ist in dos.h folgendermaßen definiert:

```
struct time {
    unsigned char ti_min;      /* Minuten */
    unsigned char ti_hour;    /* Stunden */
    unsigned char ti_hund;    /* 1/100 Sekunden */
    unsigned char ti_sec;     /* Sekunden */
};
```

Quelle: Borland C++ 5.0 Programmierhandbuch

Damit die Ergebnisse der verschiedenen Implementierungen verglichen werden können, sollen Sie die zu sortierenden Zahlen in einer Datei vorgeben. Die Datei soll zunächst eine Zahl enthalten, die angibt, wieviele Zahlen noch folgen bzw. sortiert werden sollen und dann alle diese Zahlen. Sie sollten davon ausgehen, daß bis zu 10 000 Zahlen sortiert werden sollen.

5. Aufgabe:

Nach dem Sie in der vorigen Aufgabe die Zahlen sortiert haben, ist nun ein Programm zu schreiben, daß feststellt, ob eine bestimmte Zahl in dieser Menge vorkommt. Es soll eine beliebige Zahl eingegeben werden können. Ist diese Zahl vorhanden, dann wird der Index zu der Zahl in dem Feld, in dem die Zahlen gehalten werden, ausgegeben, sonst wird eine Meldung ausgegeben, daß die Zahl nicht vorkommt.

Achten Sie wieder darauf, daß der Algorithmus effizient arbeitet