

Übungsblatt 1

Ein ganz einfacher Sortieralgorithmus auf einem Feld $a[N]$ von N Integer-Zufallszahlen ist "Selection-Sort". Dieser Algorithmus geht nach folgendem Prinzip vor:

1. Führe die folgenden beiden Schritte für $i = 0$ bis $N - 2$ aus.
2. Bestimme das Minimum $a[k]$ aller Zahlen $a[i]$ bis $a[N - 1]$
3. Tausche das Minimum $a[k]$ mit $a[i]$

Es wird also immer der kleinste Wert in einem noch nicht sortiertem Feld von Werten gesucht und an den Anfang des noch nicht sortiertem Felds gesetzt. Erzeugen Sie ein Feld $a[N]$ mit N Zufallszahlen mittels der C-Funktion `rand()`. Sortieren Sie die Zufallszahlen mit dem Selection-Sort Algorithmus.

Zählen Sie die Anzahl der Vergleiche, die Ihr Programm ausführt, und messen Sie die Rechenzeit für $N = 2^i$, $i = 10, \dots, 17$ Zahlen. Wie verhält sich die Anzahl der Vergleiche und die Rechenzeit als Funktion von N ?

Vergessen Sie nicht, Ihren Sortieralgorithmus mit wenigen Zahlen zu testen.

Hinweis:

Zeitmessungen unter Windows z.B. über

```
// zu Beginn der Datei
#include <windows.h>
...
// Am Anfang von main()
LONGLONG g_Frequency, g_FirstCount, g_LastCount;
double duration;           // gibt die Laufzeit
...
// vor dem Aufruf der Sortierfunktion
QueryPerformanceFrequency((LARGE_INTEGER*)&g_Frequency);
QueryPerformanceCounter((LARGE_INTEGER*)&g_FirstCount);
...
// nach dem Aufruf der Sortierfunktion
QueryPerformanceCounter((LARGE_INTEGER*)&g_LastCount);
duration = (((double)(g_LastCount - g_FirstCount)) / ((double)g_Frequency))
// Laufzeit duration und Größe des Feldes ausgeben
```

bitte wenden

Zeitmessungen unter Linux z.B. über

```
// zu Beginn der Datei
#include <sys/time.h>
...
// Am Anfang von main()
double start;
struct timeval tm;
...
// vor dem Aufruf der Sortierfunktion
gettimeofday(&tm, NULL);
start = (double) (tm.tv_sec) + ((double) (tm.tv_usec))/1.0e6;
...
// nach dem Aufruf der Sortierfunktion
gettimeofday(&tm, NULL);
duration = (double) (tm.tv_sec) + ((double) (tm.tv_usec))/1.0e6 - start;
// Laufzeit duration und Größe des Feldes ausgeben
```