Technische Universität München
Institut für Informatik
Prof. Dr. Ernst W. Mayr

WS 1998/99
Aufgabenblatt 10
25. Januar 1999

Praktikum Algorithmen-Entwurf

(Abgabetermin: Montag, den 1. Februar 1999, 24.^{00}Uhr)

Sortier-Algorithmen

Aufgabe 1

Es ist ein Feld a von n Strings gegeben, wobei die Strings als Zeiger auf Zeichen (`char *`) in $a[0]$, $a[1]$, ..., $a[n-1]$ gespeichert sind. Ziel ist, das Feld a so zu sortieren, daß die Reihenfolge der Strings der durch *strcmp* gegebenen lexikographischen Ordnung entspricht, d.h. in dem sortierten Feld soll für alle i von 0 bis $n-2$ gelten: $strcmp(a[i], a[i+1]) \leq 0$.

Verwenden Sie das zur Verfügung gestellte Rahmenprogramm „stringsort.C", in dem bereits das Einlesen von Strings aus einer Eingabedatei, die Zeitmessung für den Aufruf der Sortierroutinen und ein Korrektheitstest für die erzeugten Sortierungen implementiert sind. Implementieren Sie in diesem Rahmenprogramm die folgenden im Skript beschriebenen Sortier-Algorithmen:

`void bubblesort(int n, char **a);`	Bubble-Sort
`void selectionsort(int n, char **a);`	Selection-Sort
`void insertionsort(int n, char **a);`	Insertion-Sort
`void shellsort(int n, char **a);`	Shell-Sort
`void quicksort(int n, char **a);`	Quick-Sort
`void mergesort(int n, char **a);`	Merge-Sort
`void radixsort(int n, char **a);`	Radix-Sort

Das Programm „stringsort.C" erwartet als Parameter den Namen einer Textdatei. Nach dem Start liest es die Textdatei und sortiert sie mittels `qsort()` aus der Standard-Bibliothek. Danach ruft es nacheinander die obigen Sortier-Funktionen auf und gibt jeweils die Laufzeit aus und prüft, ob die Sortierung korrekt ist.

Versuchen Sie, die Algorithmen möglichst effizient zu implementieren!

Hinweise

Als Eingaben können Sie beliebige Textdateien mit maximaler Zeilenlänge 255 verwenden. Sinnvolle Eingabedateien sowie das Rahmenprogramm „stringsort.C" werden an der üblichen Stelle im WWW sowie im Verzeichnis `/usr/wiss/erlebach/praktikum/sortieren` auf den sunhalle-Rechnern zur Verfügung gestellt.