

Übung 3:

Ausgabetermin: 6. Mai 1999

Abgabe: 13. Mai 1999

Aufgabe 1: [Divide and Conquer]

Gegeben sei eine Sequenz (a_1, a_2, \dots, a_n) ganzer Zahlen. Bestimmen sie eine Teilfolge (a_i, \dots, a_j) mit $1 \leq i \leq j \leq n$, sodaß die Summe $\sum_{k=i}^j a_k$ maximal ist, d.h.,

$$\sum_{k=i}^j a_k \geq \sum_{k=i'}^{j'} a_k$$

für alle $1 \leq i' \leq j' \leq n$. Lösen Sie das Problem mittels eines *Divide & Conquer*-Ansatzes.

Aufgabe 2: [Listen]

- Geben Sie die Komplexität folgender Operationen für verschiedene Repräsentierungen von Listen an: sortiert oder unsortiert, einfach-verzeigert oder doppelt.

| Operation | unsortiert, einfach | sortiert, einfach | unsortiert, doppelt | sortiert, doppelt |
|---------------------|------------------------|----------------------|------------------------|----------------------|
| $search(L, k)$ | | | | |
| $insert(L, x)$ | | | | |
| $delete(L, x)$ | | | | |
| $successor(L, x)$ | | | | |
| $predecessor(L, x)$ | | | | |
| $minimum(L)$ | | | | |
| $maximum(L)$ | | | | |

- Eine beliebige Repräsentierung für (doppelt) verzeigerte Listen verwendet einen sogenannten "*sentinel*" oder Wächter, ein Dummy-element am Beginn der Liste. Zusätzlich verzeigert man das Ende der Liste so, daß der Nachfolger des letzten Elementes wieder die Dummy-Zelle ist (d.h., es handelt sich um eine doppelt-verzeigerte, zirkuläre Liste).

Implementieren Sie die Operationen *List-insert*, *List-delete* und *List-search* auf dieser Struktur. Argumentieren Sie, warum man dieses Dummy-Element oft einführt.