

3. Übung:

Algorithmen und Datenstrukturen

Sommersemester 2008

23. April 2008

Abgabe bis Freitag, 2. Mai 2008, vor der Vorlesung

Aufgabe 3.1: (Punkte 5)

Überlegen Sie sich eine nicht-rekursive Variante von *mergesort*.

Hinweis: Erzeugen Sie sortierte Teilfolgen, zunächst der Länge 2, dann der Länge 4, usw. bis schließlich nur noch eine Folge existiert.

Aufgabe 3.2: (Punkte 6)

Implementieren Sie doppelt verkettete Listen `int_list` für ganze Zahlen, bestehend aus Listenelementen `int_list_item`, (als Pseudo-Code oder einer Ihnen bekannten Programmiersprache) mit den Operationen:

- a) `insert(int x, int_list_item* it)`: Einfügen des Wertes `x` hinter einem gegebenen Listenelement `it`.
- b) `delete(int_list_item* it)`: Entfernen des Listenelements `it`.
- c) `concatenate(int_list* L)`: Konkatenieren zweier Listen.
- d) `split(int_list_item* it, int_list* L)`: Splitten der Liste in zwei Teillisten hinter Element `it`.

Aufgabe 3.3: (Punkte 4)

Implementieren Sie (in Pseudo-Code oder einer Programmiersprache) die dynamischen Varianten der Datenstrukturen *stack* (Keller) und *queue* (Schlange) für ganze Zahlen, d.h. die Zahl der Elemente soll beliebig sein. Benutzen Sie einfach verkettete Listen.

Aufgabe 3.4: (Punkte 5)

Überlegen Sie sich eine nicht-rekursive Variante der binären Suche.