

Algorithmen und Datenstrukturen I

WS 2009/10, **2. Aufgabenblatt**, Abgabe 11.11.2009

Aufgabe 5

10 Punkte

Gegeben sei eine lineare Liste

$$L[] = (1, 2, 3, 4, 5, 6, 7, 8, 9)$$

- (a) Behandeln Sie L als selbstorganisierende Liste mit Transpose-Regel. Suchen Sie in L nach den Schlüsseln 5, 9, 9, 5 in dieser Reihenfolge (vier separate Aufrufe der Suchroutine). Geben Sie nach jedem Suchvorgang die Anzahl der in diesem Suchvorgang ausgeführten Schlüsselvergleiche und die aktuelle Liste an.
- (b) Wie Aufgabenteil (a), jedoch mit Move-To-Front anstatt Transpose.

Aufgabe 6

9 Punkte

Gegeben sei eine lineare Liste $L[]$ der Länge $n = 25$ mit Einträgen

$$L[i] = i^2$$

für alle $i \in \{1, \dots, n\}$.

- (a) Suchen Sie nach den Schlüsseln 1, 25 und 624. Benutzen Sie dazu Interpolationssuche (vgl. Vorlesung 2, Folie 10). Geben sie zu jedem der drei Suchvorgänge die Sequenz der Listenindizes p an, an denen mit dem gesuchten Schlüssel verglichen wird. Ist $u = v$, so wird direkt $p = u$ gesetzt.
- (b) Wiederholen Sie die Suchvorgänge aus (a) unter Benutzung von Binärsuche. Geben Sie auch hier für jeden Suchvorgang die Folge der Indizes an, deren Einträge mit dem gesuchten Schlüssel verglichen werden.

Aufgabe 7

6 Punkte

Gegeben sei das folgende unsortierte Feld ganzer Zahlen:

[14, 5, 39, 7, 51, 3, 19, 2, 41, 28, 17]

Bestimmen Sie

- (a) das drittkleinste
- (b) das zehntkleinste

Element dieser Liste mit dem Verfahren aus der Vorlesung (Teil 2, Folie 19). Geben Sie hierzu in jeder Rekursion die betrachtete Teilliste an. Das pivot Element p sei jeweils der erste Eintrag der betrachteten Teilliste.

Aufgabe 8

5 Punkte

Betrachten Sie ein zwei-dimensionales Feld $L[i][j]$ mit Indizes $i, j \in \{1, \dots, m\}$, das bei Konstanthalten eines Index bezüglich des anderen Index geordnet ist.

Für alle $i, j, k, l \in \{1, \dots, m\}$ gelte also1

1. $L[i][j] \leq L[k][j]$, wenn $i < k$, und
2. $L[i][j] \leq L[i][l]$, wenn $j < l$.

Gegeben sei ein Verfahren, das ein solches Feld L und einen Schlüssel s als Eingabe erhält und feststellt, ob s in L enthalten ist. Leiten Sie eine untere Schranke der Zeitkomplexität für ein solches Verfahren her. Gibt es ein Verfahren mit Zeitaufwand in $O(\log m)$?