

Universität Leipzig Institut für Informatik Automatische Sprachverarbeitung	Algorithmen und Datenstrukturen WS 2017/18 – Serie 2		
Prof. Dr. Uwe Quasthoff	Ausgabe am 07.11.2017	Abgabe am 14.11.2013	Seite 1/2

Algorithmen und Datenstrukturen

WS 2017/18– Serie 2

Organisatorisches

Briefkastenabgaben werden nur noch bis zum Vortag der Vorlesung (Montag) 16:00 akzeptiert. Spätere Briefkastenabgaben werden als *verspätet* behandelt.

5 (6 Punkte) Suchverfahren

Gegeben sei folgende in einem Feld abgespeicherte, aufsteigend sortierte Zahlenfolge L :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$L[i]$	21	23	25	27	29	31	33	35	37	39	61	63	65	67	69	71

i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$L[i]$	72	74	76	78	90	92	94	127	129	131	133	135	137	139	141	145

Geben Sie für die Suchverfahren

- Binäre Suche
- Einfache Sprungsuche (mit optimaler Sprungweite; Kosten eines Sprunges und eines sequentiellen Vergleichs seien gleich)
- Interpolationssuche

jeweils den Suchweg bis zum Auffinden des Elements 127 an. Beschreiben sie dabei alle wesentlichen Berechnungen und Entscheidungen. Halten Sie sich strikt an das in der Vorlesung vorgestellte Vorgehen.

6 (4 Punkte) Auswahlproblem

Gegeben sei das folgende unsortierte Feld ganzer Zahlen:

[42, 7, 36, 9, 47, 3, 4, 21, 13, 10].

Bestimmen Sie

- a) das zweitkleinste
- b) das siebtkleinste

Element dieser Liste, in dem sie das für das Auswahlproblem angegebenen Divide-&-Conquer-Verfahren aus der Vorlesung anwenden; geben sie in jeder Rekursion die betrachtete Teilliste an. Benutzen sie jeweils das letzte Element der betrachteten Teilliste als Pivotelement p .

Universität Leipzig Institut für Informatik Automatische Sprachverarbeitung	Algorithmen und Datenstrukturen WS 2017/18 – Serie 2		
Prof. Dr. Uwe Quasthoff	Ausgabe am 07.11.2017	Abgabe am 14.11.2013	Seite 2/2

7 (3 Punkte) Wohlgeformte Klammersausdrücke

Geben sie Pseudocode für ein Verfahren an, welches eine beliebige Zeichenkette zeichenweise einliest und prüft, ob es sich um einen wohlgeformten Klammersausdruck handelt. Als Speicher soll das Verfahren nur einen Zähler i und einen Stack mit den Standard-Operationen PUSH, POP, TOP etc. verwenden. Nehmen sie an, dass die Zeichenkette wie üblich in einem 0-basierten Array a mit Länge n übergeben wird und für die Wohlgeformtheit nur der eckige Klammertyp $[]$ relevant ist.

8 (4 Punkte) Selbstorganisierende Listen

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 Werten $3, 7, 7, 3$ in dieser Reihenfolge (mehrere separate Aufrufe der Suchroutine; wird die Liste verändert, soll danach in der veränderten Liste weitergesucht werden). Geben Sie nach jedem Suchvorgang die Anzahl der in diesem Suchvorgang ausgeführten Vergleiche und die aktuelle Liste an.
- b) Wie Aufgabenteil (a), jedoch mit Move-To-Front anstatt Transpose. Starten sie wieder mit der ursprünglich gegebenen Liste L

9 (4 Punkte) Verkettete selbstorganisierende Listen

Eine selbstorganisierende Liste sei als verkettete Listen mit Kopfzeiger $head$ (ohne Dummy-elemente) implementiert. Jedes Element bestehe aus einem Schlüssel key , dem (für die Aufgabe nicht relevanten) Wert und einem Zeiger $next$.

Geben Sie Pseudocode für die nach erfolgreicher Suche jeweils bei der a) Transpose und b) Move-to-front-Regel benötigte Umstrukturierung der Liste an. Pseudocode für die Suche selbst müssen Sie nicht angeben. Verwenden Sie für (a) eine Schlüssel- und für (b) eine Zeiger-basierte Vorgehensweise.

Als Ergebnis der erfolgreichen Suche bei (b) zeige der Zeiger p auf das gefundene Element und der Zeiger q auf das vorhergehende Element (Ausnahme: $q = null$, falls $p = head$). Die Zeiger p , q und $head$ seien globale Variablen.

10 (4 Punkte) Sortieralgorithmen

Geben Sie für Bubble-Sort (mit Abbruchkontrolle) und Insertion-Sort an, auf welcher Art von Eingabefeldern eine maximale Anzahl an Vergleichen bzw. Vertauschungen in Abhängigkeit von der Größe n des Eingabefeldes benötigt wird. Begründen Sie jeweils ihre Antwort.