Matthias Krause
Alexander Moch

Mannheim, den 05.10.2020

CS 307 Algorithmen und Datenstrukturen, Herbstsemester 2020

Übungsblatt 2

AUFGABE 2.1: Der folgende Algorithmus benutzt die Methode der binären Suche, um festzustellen, ob ein $x \in U$ in einem (aufsteigend) sortierten (Teil-)Array $A[l \cdots r]$ von Elementen aus U vorkommt.

algorithm *BINARY-SEARCH* (A, l, r, x)
1 **if** $(l > r)$ **return** *FALSE;*
2 $m := \lfloor (l + r)/2 \rfloor$;
3 **if** $(x = A[m])$ **return** *TRUE;*
4 **if** $(x > A[m])$ **return** *BINARY-SEARCH* $(A, m + 1, r, x)$;
5 **else** **return** *BINARY-SEARCH* $(A, l, m - 1, x)$;

a) Beweisen Sie die Korrektheit von *BINARY-SEARCH*.

b) Geben Sie die maximale Anzahl der Vergleiche bei der Ausführung von *BINARY-SEARCH* $(A, 1, n, x)$ in der O-Notation an. Bei welchen Eingaben tritt der worst case ein?

c) Geben Sie eine nicht-rekursive Realisierung der binären Suche in Pseudocode an.

AUFGABE 2.2: Gegeben sind ein Array $A[1 \cdots n]$ von Integers und ein Integer y. Zu entscheiden ist, ob es Indizes i und j, $1 \leq i \leq j \leq n$, gibt mit $y = A[i] + A[j]$. Beschreiben Sie, wie man dieses Problem in Zeit $O(n \log n)$ und Platz $\Theta(n)$ lösen kann.

AUFGABE 2.3: Schreiben Sie ein Pseudocodeprogramm für die Operation Merge(A, p, s, q). Hierbei ist A ein Feld mit Einträgen aus einer geordneten Menge $(M, \leq)$. Es gilt $1 \leq p \leq s \leq q \leq \text{length}(A)$. Merge$(A, p, s, q)$ sortiert, unter Nutzung eines Hilfsfeldes B, in Linearzeit $O(q - p)$ das Teilfeld $A[p], \ldots, A[q]$ unter der Bedingung, dass die Teilfelder $A[p], \ldots, A[s]$ und $A[s + 1], \ldots, A[q]$ bereits sortiert sind.

AUFGABE 2.4: Man bestimme mit der Summations-Integrationsmethode eine möglichst genaue untere und obere Schranke für $\sum_{k=1}^{n} k^2 \ln(k)$.