

Staatsexamen 46115 / 2021 / Frühjahr / Thema Nr. 1 / Teilaufgabe Nr. 2 / Aufgabe Nr. 1

Aufgabe

Aufgabe 1 (Sortieren) [26 PUNKTE]

a) Geben Sie für folgende Sortierverfahren jeweils zwei Felder A und B an, so dass das jeweilige Sortierverfahren angewendet auf A seine Best-Case-Laufzeit und angewendet auf B seine Worst-Case-Laufzeit erreicht. (Wir messen die Laufzeit durch die Anzahl der Vergleiche zwischen Elementen der Eingabe.) Dabei soll das Feld A die Zahlen 1,2,...,7 genau einmal enthalten; das Feld B ebenso. Sie bestimmen also nur die Reihenfolge der Zahlen.

Wenden Sie als Beleg für Ihre Aussagen das jeweilige Sortierverfahren auf die Felder A und B an und geben Sie nach jedem größeren Schritt des Algorithmus den Inhalt der Felder an.

Geben Sie außerdem für jedes Verfahren asymptotische Best- und Worst-Case-Laufzeit für ein Feld der Länge n an.

Für drei der Sortierverfahren ist der Pseudocode angegeben. Beachten Sie, dass die Feldindizes hier bei 1 beginnen. Die im Pseudocode verwendete Unterroutine $\text{Swap}(A, i, j)$ vertauscht im Feld A die Elemente mit den Indizes i und j miteinander.

i) Insertionsort ii) Bubblesort iii) Quicksort

Insertionsort(int[] A) for $i = 2$ to A.length do key = A[i] $i = j - 1$ while $i > 0$ and A[i] > key do A[i + 1] = A[i]

$i = i + 1$ A[i] = key
Bubblesort(int[] A) n := length(A) repeat swapped = false for $i = 1$ to n - 1 do if A[i] > A[i + 1] then Swap(A, i, i + 1)

swapped := true
until not swapped
Quicksort(int[] A, l = 1, r = A.length) if $2 < r$ then m = Partition(A, l, r) | Quicksort(A, l, m - 1) Quicksort(A, m + 1, r)

int Partition (int[] A, int l, int r)

pivot = A[r]

i = l

for j = l to r - 1 do

if A[j] < pivot then

Swap(A, i, j) i = i + 1

Swap(A, i, r)

return i

b) Geben Sie die asymptotische Best- und Worst-Case-Laufzeit von Mergesort an.

Github: Staatsexamen/46115/2021/03/Thema-1/Teilaufgabe-2/Aufgabe-1.tex