

Übungen zur Vorlesung Grundlagen der Programmierung II Blatt 12

Aufgabe 1:

Bestimmen Sie sechs Algorithmen, die zu N ganzen Zahlen und gegebenem $K, T \in \mathbb{N}$ berechnen, ob es unter den N Zahlen K Zahlen gibt, deren Summe höchstens T beträgt. Die Algorithmen sollen folgende Laufzeiten besitzen:

$O(N^K)$, $O(N^2)$, $O(N \cdot K)$, $O(N \cdot \log N)$, $O(N \cdot \log K)$, $O(N + K \cdot \log N)$

Aufgabe 2:

Für die folgende Schlüsselfolge

51 9 53 7 91 17 90 28 66 43 16 52 62 68 77 71

bestimme man den Ausgangsheap sowie die Heaps nach den ersten drei Schritten des Heapsort-Algorithmus. Man benutze die Darstellung durch Binärbäume.

Aufgabe 3:

In einem Array der Größe n seien nur die Werte rot, blau, grün gespeichert. Das Array soll so sortiert werden, daß zunächst alle roten, dann alle grünen und zuletzt alle blauen Farben vorkommen. Die einzigen erlaubten Zugriffsoperationen auf das Array sind

$\text{test}(i)$: Liefert die Farbe der i -ten Komponente

$\text{tausch}(i,j)$: tauscht die Inhalte der beiden Komponenten i und j

Schreiben Sie einen Algorithmus, der das Array in situ in $O(n)$ Zeit sortiert. Zeigen Sie, daß Ihr Algorithmus in $O(n)$ Schritten terminiert und daß der Array dann tatsächlich sortiert ist.