

Einsendeaufgabe 4

Stefan Berger

1.

Gegeben sei das Zahlenfeld $A = (11, 15, 13, 8, 1, 10, 16, 12)$. A ist nach dem Algorithmus Quicksort aufsteigend zu sortieren.

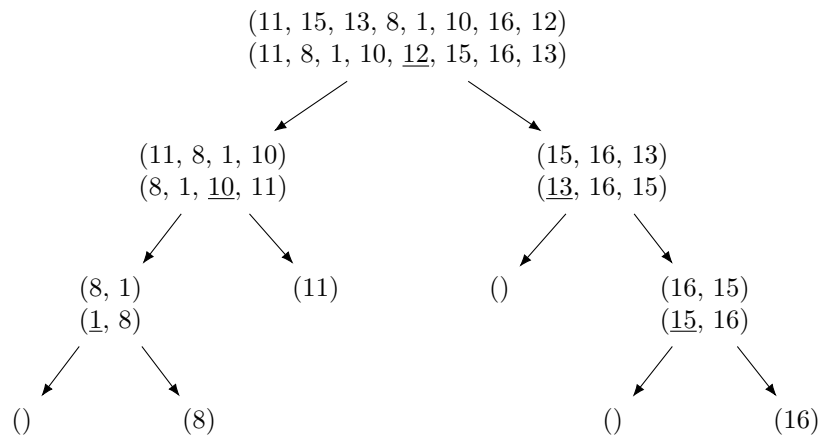
(a) Führen Sie die erste Teilung nach dem im Skript gegebenen Verfahren durch.

i. Geben Sie die dabei auftretenden Vertauschungen der Reihe nach an.

ii. Welches ist der erste Teilungsindex?

p	r	$A[1..6]$	x	j	$A[j] \leq x$	i	$A[i] \leftrightarrow A[j]$	$A[i] \leftrightarrow A[r]$
1	8	(11, 15, 13, 8, 1, 10, 16, 12)	12	1	$11 \leq 12$	1	$11 \leftrightarrow 11$	
		(11, 15, 13, 8, 1, 10, 16, 12)		2	$15 \leq 12$			
				3	$13 \leq 12$			
				4	$8 \leq 12$	2	$15 \leftrightarrow 8$	
		(11, 8, 13, 15, 1, 10, 16, 12)		5	$1 \leq 12$	3	$13 \leftrightarrow 1$	
		(11, 8, 1, 15, 13, 10, 16, 12)		6	$10 \leq 12$	4	$15 \leftrightarrow 10$	
		(11, 8, 1, 10, 13, 15, 16, 12)		7	$16 \leq 12$			
						5		$13 \leftrightarrow 12$
		(11, 8, 1, 10, 12, 15, 16, 13)						

(b) Stellen Sie den gesamten Ablauf in einem Binärbaum dar, so dass insbesondere das rekursive Teilen des Zahlenfeldes deutlich wird. Markieren Sie darin auch die jeweiligen Pivotelemente.



3.

Testen Sie Ihre Methode mit den Zahlen in der Datei QuickSort.txt Copyright Tim Roughgarden, Stanford University. Dafür benutzen Sie die Klasse UtilitiesArraysFiles. Wie viele Vergleiche bekommen Sie? Passt das Ergebnis mit dem theoretischen Ergebnis im besten Fall $n \log n$? (Log sollte mit Basis 2 berechnet werden)

Im besten Fall benötigt der Algorithmus $n \log n$ Vergleiche, z.B. $10000 \cdot \log_2(10000) = 132877.1238$. Im Beispiel mit QuickSort.txt führt die Implementierung 160361 Vergleiche durch, das ist in dem Bereich zwischen dem besten Fall $T(n) \in \Omega(n \log n)$ und dem schlechtesten Fall $T(n) \in \mathcal{O}(n)$.

4.

Stellen Sie die Rekursionsgleichung zur Bestimmung der Zeitkomplexität vom Quicksort im idealen Fall auf: das Array wird immer in zwei geteilt.

Der Verzweigungsfaktor beträgt $2 \rightarrow b = 2$

Das Array wird mit jedem Rekursionsschritt durch 2 geteilt $\rightarrow c = 2$.

Die Komplexität von $f(n)$ ist $T(n) = \Theta(n - 1)$

Die Rekursionsgleichung lautet daher $2T(n/2) + f(n)$ mit $f(n) \in \Theta(n)$.

5.

Lösen Sie die Rekursionsgleichung mit Hilfe des Master-Theorems und geben sie die Zeit- Komplexität in asymptotischer Notation.

Der Exponent k von n in $\Theta(n^k)$ ist für $f(n) \in \Theta(n)$ gleich $1 \rightarrow k = 1$

Die Lösung der Rekursionsgleichung mithilfe des Master-Theorems lautet daher:

$$T(n) \in \Theta(n^k \log n), \text{ falls } b(1/c)^k = 1$$

$$b = 2, c = 2, k = 1$$

$$2(1/2)^1 = 1$$

$$T(n) \in \Theta(n \log n)$$