Der Rest

Wie sieht unser Array $a[1, \ldots, n]$ nun aus?

Durch die beschriebene Prozedur war zunächst das kleinste der $a[i]$ auf Platz $a[n]$ getauscht worden, dann das Zweitkleinste auf Platz $a[n-1]$ usw.

Machen Sie sich bitte selbst klar, dass dabei nie eines der im Quicksort-Schritt in der zweiten Hälfte des Arrays gelandeten Elemente (also derer, die größer als der Median sind), an die Position $a[1]$ gelangen kann.

Es liegen nun also in $a[1, \ldots, \lfloor n/2 \rfloor]$ die noch unsortierten Array-Elemente, die allesamt größer als der Median sind. Der übrige Teil ist bereits (von groß nach klein) sortiert.

Nun rufen wir rekursiv das Programm wieder auf, und zwar mit der Eingabe $a[1, \ldots, \lfloor n/2 \rfloor]$.

Analyse

Wir gehen von einem Algorithmus zur Medianberechnung aus, der mit einer linearen Anzahl Vergleiche auskommt. Der anschließende Quicksortschritt und der Heapaufbau auf der ersten Hälfte des Arrays brauchen ebenfalls maximal linear viele Vergleiche, so dass wir höchstens $c \cdot n$ (für geeignetes c) Vergleiche verbraucht haben, bis wir mit dem *Herauspflücken* beginnen.

Wir sortieren $n/2$ Elemente ein, wobei das getauschte Element ziemlich genau aus $\log n$ Höhe bis ganz nach unten einsinkt.

Man beachte: Dabei wird der Heap jedesmal um ein Element kleiner!

Zum Schluss ruft sich die Prozedur rekursiv mit halb so großem Eingabe-Array auf. Das bedeutet:

$$T(n) = c \cdot n + \tfrac{n}{2} \log n + T(\tfrac{n}{2})$$

Diese Rekursionsformel hat offenbar die Lösung

$$T(n) = n \log n + 2c \cdot n.$$

Ein Beispiel

Wir wählen $n = 31$ und der Einfachheit halber seien die zu sortierenden Array-Elemente gerade die Zahlen 1 bis 31. Der Median ist dann 16, und die Aufteilung nach dem Quicksort-Schritt sei wie folgt (grün: $a[1]$ bis $a[16]$, blau: $a[17]$ bis $a[31]$)

5 15 10 1 4 8 16 11 7 2 6 3 13 12 9 14
25 26 28 17 31 30 23 22 18 29 19 20 27 24 21

Als nächstes wird der grüne Teil zum Heap gemacht. Ergebnis:

1 2 3 7 4 8 9 11 15 5 6 10 13 12 16 14
25 26 28 17 31 30 23 22 18 29 19 20 27 24 21

Nun 1 mit 21 tauschen und 21 ganz nach unten sinken lassen:

2 4 3 7 5 8 9 11 15 17 6 10 13 12 16 14
25 26 28 21 31 30 23 22 18 29 19 20 27 24 1

Jetzt 2 mit 24 tauschen, Einsinkelemente sind 3, 8, 10, 22.

Beispiel (Forts.)

3 4 8 7 5 10 9 11 15 17 6 22 13 12 16 14
25 26 28 21 31 30 23 24 18 29 19 20 27 2 1

Jetzt 3 mit 27 tauschen, Einsinkelemente sind 4, 5, 6, 30.

4 5 8 7 6 10 9 11 15 17 30 22 13 12 16 14
25 26 28 21 31 27 23 24 18 29 19 20 3 2 1

Jetzt 4 mit 20 tauschen, Einsinkelemente sind 5, 6, 17, 21.

5 6 8 7 17 10 9 11 15 21 30 22 13 12 16 14
25 26 28 20 31 27 23 24 18 29 19 4 3 2 1

Nun machen wir einen Zeitsprung um 12 weitere Schritte. Die „blauen“Elemente (17 bis 31) liegen (unsortiert) in den ersten Komponenten, also als $a[1, \ldots, 15]$, der Rest des Arrays sieht genau so aus:

16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Den Rest erledigt der rekursive Aufruf.

Medianberechnung in Linearzeit

Um den Median von n Array-Elementen zu berechnen, d.h. den Index r zu finden, für den $|\{i \mid a[i] \leq a[r]\}| \geq n/2$ und $|\{i \mid a[i] \geq a[r]\}| \geq n/2$ gilt, sollen wir nur $O(n)$ Vergleiche durchführen. Dazu müssen einige Tricks angewendet werden.

Wie so oft, benötigen wir auch hier zunächst ein nachweislich gutes Pivot-Element, um eine Aufteilung in zwei annähernd gleich große Teile zu erreichen.

Nach der Aufteilung müssen wir nicht in beiden Teilen weiter suchen, denn der Median kann bei einer Aufteilung, bei der der erste Teil größer als der zweite ist, nur im ersten Teil liegen, und ebenso im anderen Fall nur im zweiten Teil.

Allerdings brauchen wir jetzt nicht mehr den Median, sondern das k-kleinste Element für ein bestimmtes $k \in \{1, \ldots, \lceil n/2 \rceil\}$.

Erster Schritt: Median aus 5

Zunächst suchen wir ein gutes Pivot-Element.

Wir berechnen aus je 5 Elementen einen *lokalen* Median.
Dazu braucht man 6 Vergleiche:

Die Elemente seien a, b, c, d, e.

IF $a > b$ THEN vertausche(a, b) ENDIF;

IF $c > d$ THEN vertausche(c, d) ENDIF;

IF $a > c$ THEN vertausche(a, c); vertausche(b, d) ENDIF;

Nun ist a kleiner als b, c und d, daher nicht der Median.

IF $b > e$ THEN vertausche(b, e) ENDIF;

IF $b > c$ THEN vertausche(b, c); vertausche(e, d) ENDIF;

Nun ist b kleiner als c, d und e, daher nicht der Median.
Außerdem ist d größer als a, b und c, also auch nicht der Median.
Ein Vergleich zwischen c und e genügt nun zur Ermittlung des gesuchten Medians. (Das kleinere von beiden ist der Median!)