

Übungen zur Vorlesung  
**Datenstrukturen, Algorithmen und Programmierung 2 (DAP2)**  
Sommersemester 2007  
Blatt 9

**Aufgabe 9.1 (5 Punkte)**

Der in der Vorlesung vorgestellte Quicksort-Sortieralgorithmus verwendet eine Hilfsfunktion `partition` (Skript, S.99), welche eine gegebene Eingabefolge in zwei Teilfolgen  $a_1, \dots, a_{p-1}$  sowie  $a_{p+1}, \dots, a_n$  zerlegt.  $a_p$  enthält dann das Zerlegungsdatum, und es gilt  $a_i \leq a_p$ ,  $1 \leq i \leq p-1$ , sowie  $a_i \geq a_p$ ,  $p+1 \leq i \leq n$ . Wende diese Hilfsfunktion `partition` aus dem Skript auf die folgende Eingabe an:

11, 5, 29, 83, 28, 23, 15, 27, 56, 42, 24, 82, 76, 94.

Als Zerlegungsdatum soll das achte Element (27) dienen. Gib nach jedem Schritt die Belegung des Arrays sowie die Positionen für  $i$ ,  $l$  und  $r$  an.

**Aufgabe 9.2 (5 Punkte)**

Wende den Heapsort-Algorithmus aus der Vorlesung (Algorithmus 4.5.2, Skript S. 104) mit Strategie 1 auf folgende Eingabe an:

40, 15, 31, 8, 2, 6, 22, 12

Zeichne dabei die Zustände des Heaps nach jedem Aufruf von `reheap` in der Creation- und Selection-Phase auf.

**Aufgabe 9.3 (5 Punkte)**

Zeichne die ersten 5 Ebenen des Entscheidungsbaums für Heapsort mit linearer bottom-up Suche bei Eingabelänge  $n = 4$ . Verzichte dabei auf die Auflistung der jeweils möglichen Permutationen, sondern markiere die Knoten nur mit den Vergleichen  $(i, j)$ .

Hinweis: Es geht einfacher, wenn du den jeweils aktuellen Heap nach einer Änderung an die entsprechende Stelle des Baums zeichnest und durch Linien markierst, welche Teile des Baums den einzelnen `reheap`-Aufrufen entsprechen.

**Aufgabe 9.4 (5 Punkte)**

Berechne die Anzahl wesentlicher Vergleiche für Quickselect für  $n = 5$  und  $k = 3$  sowohl für den worst case als auch für den best case, unter der Einschränkung, dass das gesuchte Datum niemals als Zerlegungsdatum gewählt wird.