

## 1) Eigenschaften von Heaps

Treffen folgende Aussagen zu? Warum? Warum nicht?

	Ja	Nein	Begründung
a Heaps kann man als spezielle binäre Suchbäume ansehen, bei denen alle Stufen bis auf die letzte vollständig gefüllt sind und zwischen den Vätern und Söhnen des Baums die Heapordnung erfüllt ist. Die Heapordnung ist z.B., dass der Schlüssel des Vaters größer oder gleich ist wie die Schlüssel der Söhne.		X	kein Suchbaum
b Das Einfügen in einen Heap kann man zerlegen in die Schritte: Hinten an Heap anfügen, Heapordnung durch downHeap wiederherstellen.		X	nicht downHeap, sondern upHeap
c Das Entnehmen des größten Elements kann man zerlegen in die Schritte: erstes Element entnehmen und letztes Element an erste Stelle verschieben, Heapordnung wiederherstellen.	X		
d Die Operationen Einfügen eines Elements, Löschen größten Elements sowie Löschen eines beliebigen Elements haben eine Laufzeit von $O(\log n)$ .		X	Nur Löschen des größten Elements implementiert

## 2) Einfügen und Löschen in Heaps

Gegeben ist folgendes Array:

0	1	2	3	4	5	6	7
	50	44	25	33	21	6	33

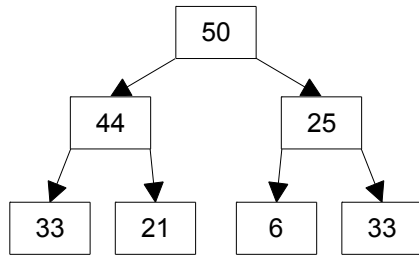
- Stellt das Array einen Heap (höherer numerischer Wert ist höhere Priorität) dar? Wenn es keinen Heap darstellt, strukturieren Sie das Array so um, dass es einen Heap darstellt. Zeichnen Sie den Heap anschließend als Binärbaum.
- Führen Sie die folgenden Operationen durch (E = Einfügen eines Elements mit gegebener Nummer, L = Löschen des höchstpriorien Elements):

E60, E12, E22, L, E20, E45, L, E21, L, L, L

Zeichnen Sie den Heap (als Array oder Graph) nach jedem Schritt und kennzeichnen Sie, welche Vertauschoperationen im Heap beim Einfügen oder Löschen erfolgen.

**Lösung:**

Als Binärbaum:



Ist kein Heap, weil der rechte Sohn von 25 größer ist als sein Vater.

**Vorgehensweise beim Einfügen:**

Als letztes Element einfügen und nach oben/vorne wandern lassen, solange es wichtiger als sein Vater ist.

**Vorgehensweise beim Löschen:**

Letztes Element an Eintrag Nummer 1 stellen und nach unten/hinten wandern lassen, bis es mindestens gleich wichtig ist wie seine Söhne. Beim nach unten/hinten Wandern lassen soll man den Eintrag mit dem wichtigeren Sohn vertauschen.

Nach Vertauschen von 25 und 33 ist es ein Heap:

0	1	2	3	4	5	6	7
	50	44	33	33	21	6	25

E60:

0	1	2	3	4	5	6	7	8
	50	44	33	33	21	6	25	60
0	1	2	3	4	5	6	7	8
	<b>60</b>	<b>50</b>	33	<b>44</b>	21	6	25	<b>33</b>

E12:

0	1	2	3	4	5	6	7	8	9
	60	50	33	44	21	6	25	33	12

E22:

0	1	2	3	4	5	6	7	8	9	10
	60	50	33	44	21	6	25	33	12	22
0	1	2	3	4	5	6	7	8	9	10
	60	50	33	44	<b>22</b>	6	25	33	12	<b>21</b>

L:

0	1	2	3	4	5	6	7	8	9
	21	50	33	44	22	6	25	33	12
0	1	2	3	4	5	6	7	8	9
	<b>50</b>	<b>44</b>	33	<b>33</b>	22	6	25	<b>21</b>	12

E20:

0	1	2	3	4	5	6	7	8	9	10
	55	44	33	33	22	6	25	21	12	20

E45:

0	1	2	3	4	5	6	7	8	9	10	11
	55	44	33	33	22	6	25	21	12	20	45
0	1	2	3	4	5	6	7	8	9	10	11
	55	<b>45</b>	33	33	<b>44</b>	6	25	21	12	20	<b>22</b>

L:

0	1	2	3	4	5	6	7	8	9	10
	22	45	33	33	44	6	25	21	12	20
0	1	2	3	4	5	6	7	8	9	10
	<b>45</b>	<b>44</b>	33	33	<b>22</b>	6	25	21	12	20

E21:

0	1	2	3	4	5	6	7	8	9	10	11
	45	44	33	33	22	6	25	21	12	20	21

L:

0	1	2	3	4	5	6	7	8	9	10
	21	44	33	33	22	6	25	21	12	20
0	1	2	3	4	5	6	7	8	9	10
	<b>44</b>	<b>33</b>	33	<b>21</b>	22	6	25	21	12	20

L:

0	1	2	3	4	5	6	7	8	9
	20	33	33	21	22	6	25	21	12
0	1	2	3	4	5	6	7	8	9
	<b>33</b>	<b>22</b>	33	21	<b>20</b>	6	25	21	12

L

0	1	2	3	4	5	6	7	8
	12	22	33	21	20	6	25	21
0	1	2	3	4	5	6	7	8
	<b>33</b>	22	<b>25</b>	21	20	6	<b>12</b>	21