Data Structures and Algorithms

Cecilia Y. Sui

October 15, 2018

Ex.1: Suppose that the sequence T H I * S * * I S A * P R I O * R * * I * T * Y * * * Q U E * * * U * E (where a letter means insert and an asterisk mean remove the maximum) is applied to an initially empty priority queue. Give the sequence of letters returned by the remove the maximum operations.

End result: H A E E

Sequence of Removed Letters: **T S I S R R P O T Y I I U Q I U**

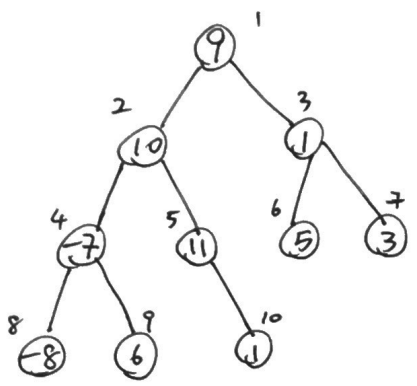
Ex.2: Sort the following array using heapsort discussed in class (page 324). Show the contents of the array and the corresponding heap tree for each step.

int [] array = {9, 10, 1, -7, 11, 5, 3, -8, 6, 1}

N	k	0	1	2	3	4	5	6	7	8	9	10
Initial values			9	10	1	-7	11	5	3	-8	6	1
10	5		9	10	1	-7	**11**	5	3	-8	6	1
10	4		9	10	1	**6**	11	5	3	-8	-7	1
10	3		9	10	**5**	6	11	1	3	-8	-7	1
10	2		9	**11**	5	6	10	1	3	-8	-7	1
10	1		**11**	10	5	6	9	1	3	-8	-7	1
Heap-ordered			11	10	5	6	9	1	3	-8	-7	1
9	1		10	9	5	6	1	1	3	-8	-7	**11**
8	1		9	6	5	-7	1	1	3	-8	**10**	11
7	1		6	1	5	-7	-8	1	3	**9**	10	11
6	1		5	1	3	-7	-8	1	**6**	9	10	11
5	1		3	1	1	-7	-8	**5**	6	9	10	11
4	1		1	-7	1	-8	**3**	5	6	9	10	11
3	1		1	-7	-8	**1**	3	5	6	9	10	11
2	1		-7	-8	**1**	1	3	5	6	9	10	11
1	1		-8	**-7**	1	1	3	5	6	9	10	11
Sorted result			-8	-7	1	1	3	5	6	9	10	11

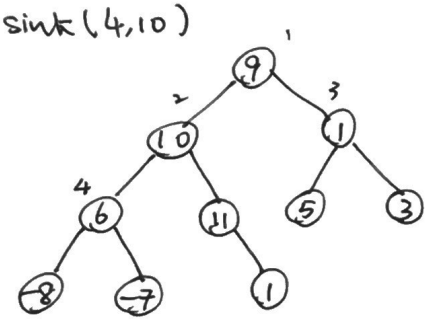
Please see next page for the corresponding heap trees.

heap construction

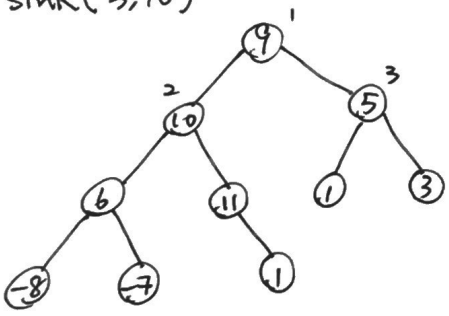


sink(5,10) – same heap tree

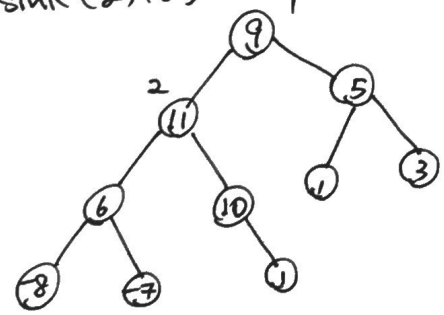
sink(4,10)



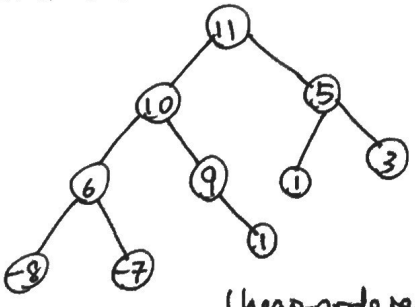
sink(3,10)



sink(2,10)

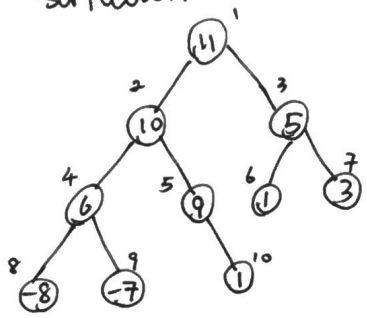


sink(1,10)

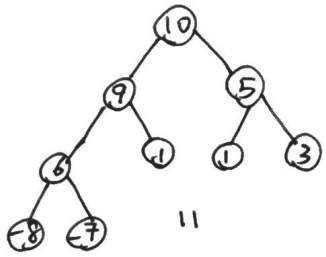


(heap-ordered)

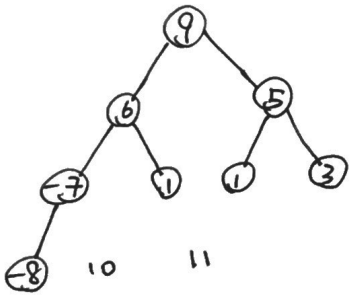
sortdown



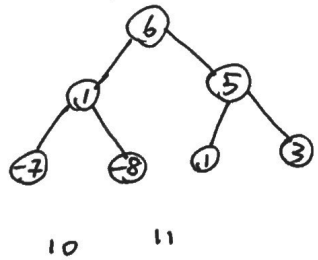
exch(1,10)
sink(1,9)



exch(1,9)
sink(1,8)

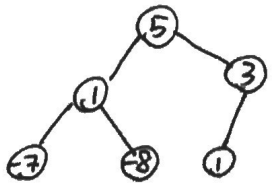


exch(1,8)
sink(1,7)



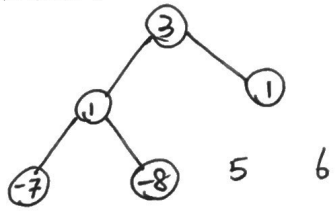
9 10 11

exch(1,7)
sink(1,6)



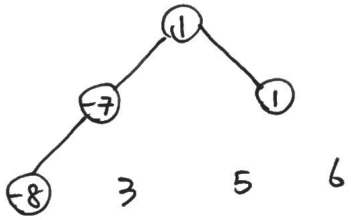
9 10 11

exch(1,6)
sink(1,5)



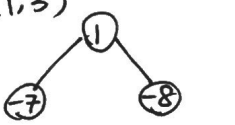
9 10 11

exch(1,5)
sink(1,4)



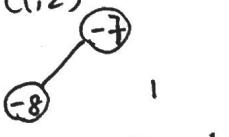
9 10 11

exch(1,4)
sink(1,3)



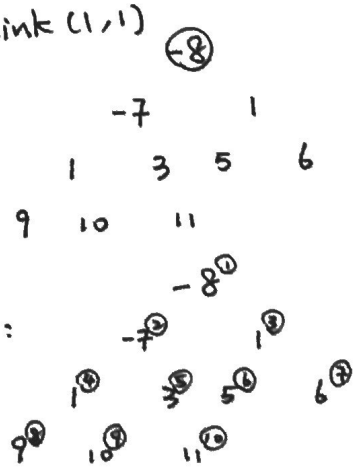
9 10 11

exch(1,3)
sink(1,2)



9 10 11

exch(1,2)
sink(1,1)



result: