Worksheet: Heaps

Worksheet: Heap Practice Group 11

In Preparation: Read Chapter 11 on the Priority Queue ADT and Heaps Insert the following values, in the order that they are given into a Min Heap. Show the tree after each insertion.

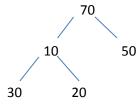
1) 30,20,50,10,5,70

30,20,30,10,3,70	T
30 is added	30
20 is added	Element is added
Tree is filled from left to right	30
New element is placed in the next available	
position, then the tree is fixed by percolating up	20
, , , , , , , , , , , , , , , , , , , ,	
(while new element is less than parent, swap value	Percolating up
with the parent)	
	30′
50 is added	20
Tree is filled from left to right	
	30 50
10 is added	Element is added
Tree is filled from left to right	20
New element is placed in the next available	/ \
position, then the tree is fixed by percolating up	30 50
(while new element is less than parent, swap value	
with the parent)	10
with the parenty	
	Percolating up
	10
	20′ 50
	30
5 is added	Element is added
Tree is filled from left to right	10
New element is placed in the next available	
position, then the tree is fixed by percolating up	20 50
(while new element is less than parent, swap value	
with the parent)	30 5
' '	Percolating up
	U-I
	5
	10 50
	10 30
	20
	30 20
70:	
70 is added	5
	10/
	10 50
	30 20 70

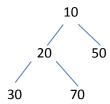
2) Remove Min from the heap

Remove Min is equivalent to removing root

Root is replaced with the element in the last filled position

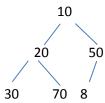


The heap is fixed by percolating down (while greater than the smallest child, swap with the smallest child)

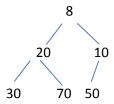


3) Add 8 to the heap

Tree is filled from left to right



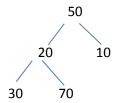
New element is placed in the next available position, then the tree is fixed by percolating up (while new element is less than parent, swap value with the parent)



4) Remove Min from the heap

Remove Min is equivalent to removing root

Root is replaced with the element in the last filled position



The heap is fixed by percolating down (while greater than the smallest child, swap with the smallest child)

