Kumal Patel

CSE 310 Recitation 4

Objectives:

1. Heaps (array representation, tree representation, max heap, min heap), Max-heapify (heap function).

Instruction

1. For all recitation: **the solution should be clearly typed or written and must be saved in .pdf or .jpg format**. Note: **unreadable answer receives no credits!**
2. All recitation must be submitted through the link posted on Blackboard, we do NOT accept any hand-in submissions or submissions sent through emails!

Question

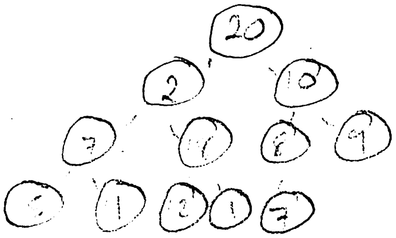
1. For the BUILD-MAX-HEAP(A) procedure we discussed in class (pseudo code given below), can we change the code in line 2, to increase the loop index *i* from 1 to length[A]/2? Why or why not?

```
BUILD-MAX-HEAP(A)
1   heap-size[A] = length[A]
2   for i = ⌊length[A]/2⌋ to 1 do
3           HEAPIFY(A, i)
```

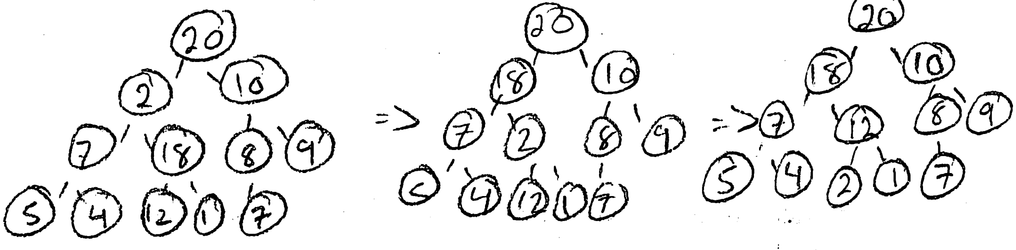
No, in order to build max heap, you need to start in the middle. Because its a recursive call of heapify, and heapify swaps the children of a parent node to let the largest be the parent. And if the bottom of the tree is not a max heap, then you won't be able to make one if you start at the top.

2. Consider the integer array $A = [20, 2, 10, 7, 18, 8, 9, 5, 4, 12, 1, 7]$

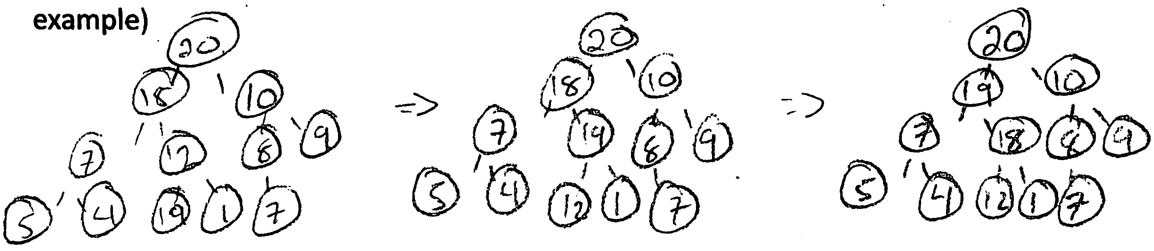
A. Give the complete binary tree representation for the array A



B. You can see that *A* is not a max heap. Use the pseudo code BUILD-MAX-HEAP(A) (textbook pp. 157) to draw the intermediate complete binary trees until you get a max heap (see pp.158 Figure 6.3 as an example).



C. HEAP-INCREASE-KEY(*A, i, key*) is an operation used in priority queues and we discussed it in class. Given the max-heap you get in above step B, show the operations of HEAP-INCREASE-KEY(*A, 10, 19*) by drawing the intermedia binary tree step-by-step (see pp.165 Figure 6.5 as one example)


20
18
10
7
12
8
9
5
4
19
1
7
=>
20
18
10
7
19
8
9
5
4
12
1
7
=>
20
19
10
7
18
8
9
5
4
12
1
7