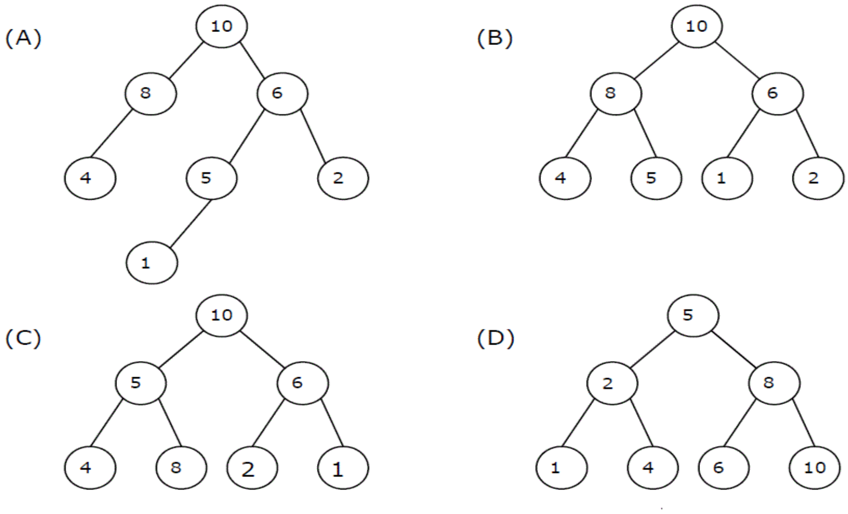
Handwriting Assignment #2 Solution

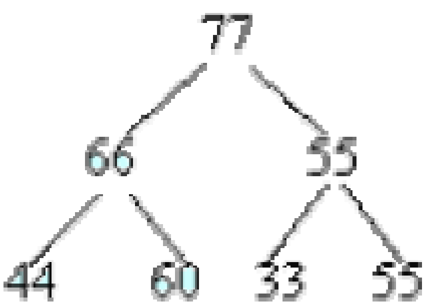
1. Which of the following is a max-heap?



Solution) (B)

2. Draw the following list of numbers as a heap with the first number as the root: 77, 66, 55, 44, 60, 33, 55

solution)



3. What is the minimum and maximum numbers of elements in a heap of height h?

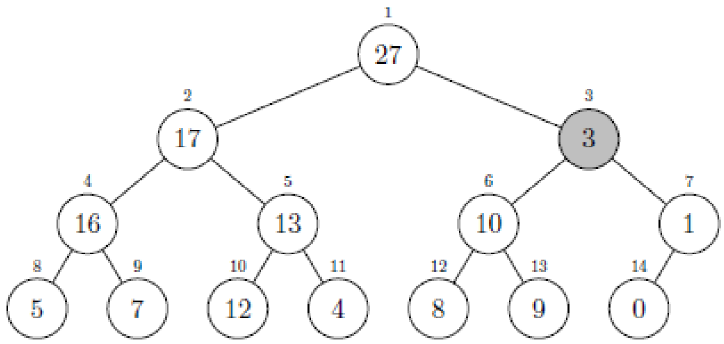
solution)

Minimum # of nodes happens in a heap in which the last level contains only one node.
Thus, minimum # of nodes $= 2^0 + 2^1 + 2^2 + \ldots 2^{h-1} + 1 = 2^h$

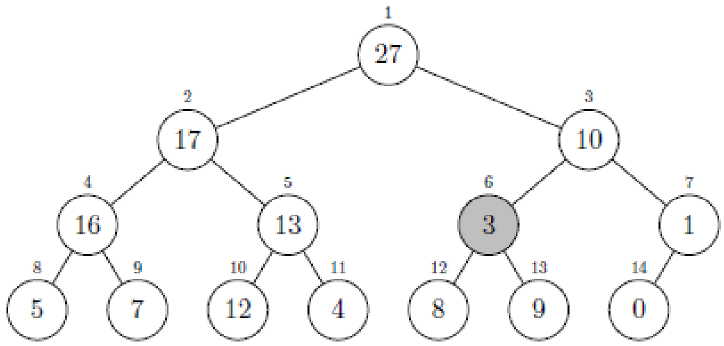
Maximum # of nodes happens in a heap in which the last level is full.
Thus, maximum # of nodes $= 2^0 + 2^1 + 2^2 + \ldots 2^h = 2^{h+1} - 1$.

4. Illustrate the operation of Max-Heapify(A, 3) on the array
   A = <27, 17, 3, 16, 13, 10, 1, 5, 7, 12, 4, 8, 9, 0>.
   solution)

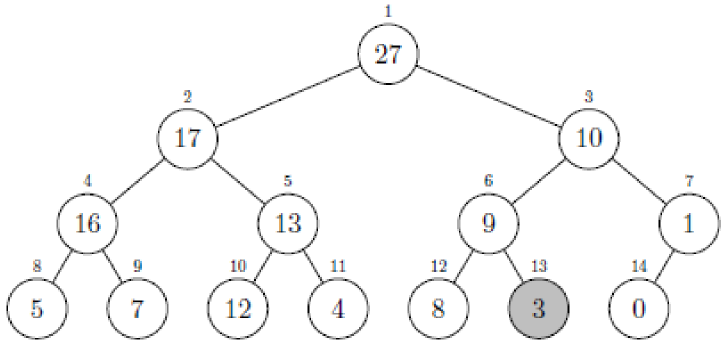
Step 0:



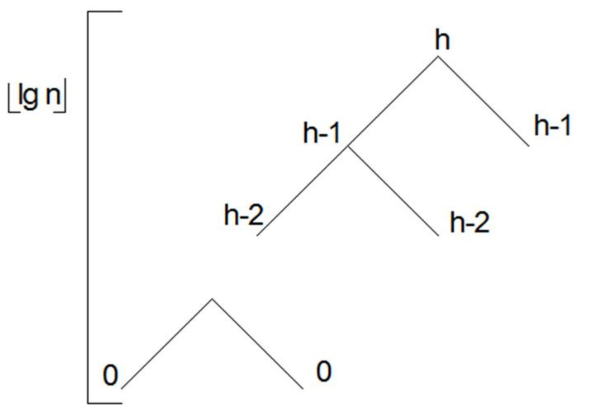
Step 1:



Step 2:



5. Build Heap is used to build a max(or min) binary heap from a given array. Build Heap is used in Heap Sort as a first step for sorting. What is the time complexity of Build Heap operation?



Height	Nodes	Cost
h	2^0	$h*2^0$
h-1	2^1	$(h-1)*2^1$
h-2	2^2	$(h-2)*2^2$
0	2^h	$0*2^h$

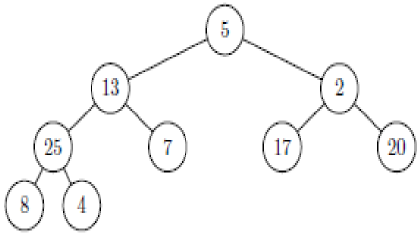
$$\sum_{h=0}^{\lfloor \lg n \rfloor} h * 2^{\lfloor \lg n \rfloor - h} = \sum_{h=0}^{\lfloor \lg n \rfloor} h * \frac{2^{\lfloor \lg n \rfloor}}{2^h} \le \sum_{h=0}^{\lfloor \lg n \rfloor} h * \frac{2^{\lg n}}{2^h} = \sum_{h=0}^{\lfloor \lg n \rfloor} h * \frac{n^{\lg 2}}{2^h}$$

$$= \sum_{h=0}^{\lfloor \lg n \rfloor} h * \frac{n}{2^h} = n \sum_{h=0}^{\lfloor \lg n \rfloor} \frac{h}{2^h} \le n \sum_{h=0}^{\infty} \frac{h}{2^h} = 2n = O(n)$$

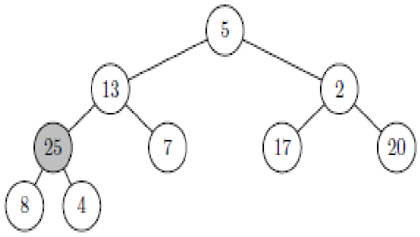
6. Illustrate the operation of Heapsort on the array
   A = <5, 13, 2, 25, 7, 17, 20, 8, 4>.
   solution)

Since $A.length = 9$, the command MAX-HEAPIFY(A, i) is called for $i = 4, 3, 2, 1$. The action of BUILD-MAX-HEAP is as follows (these first few diagrams are not required for a correct answer), with the nodes exchanged at each step shaded:

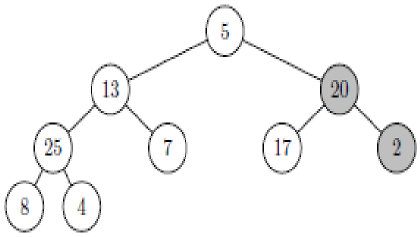
BUILD-MAX-HEAP(A):



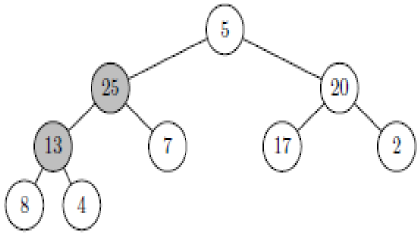
MAX-HEAPIFY$(A, 4)$:



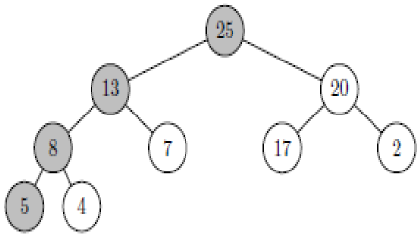
MAX-HEAPIFY$(A, 3)$:

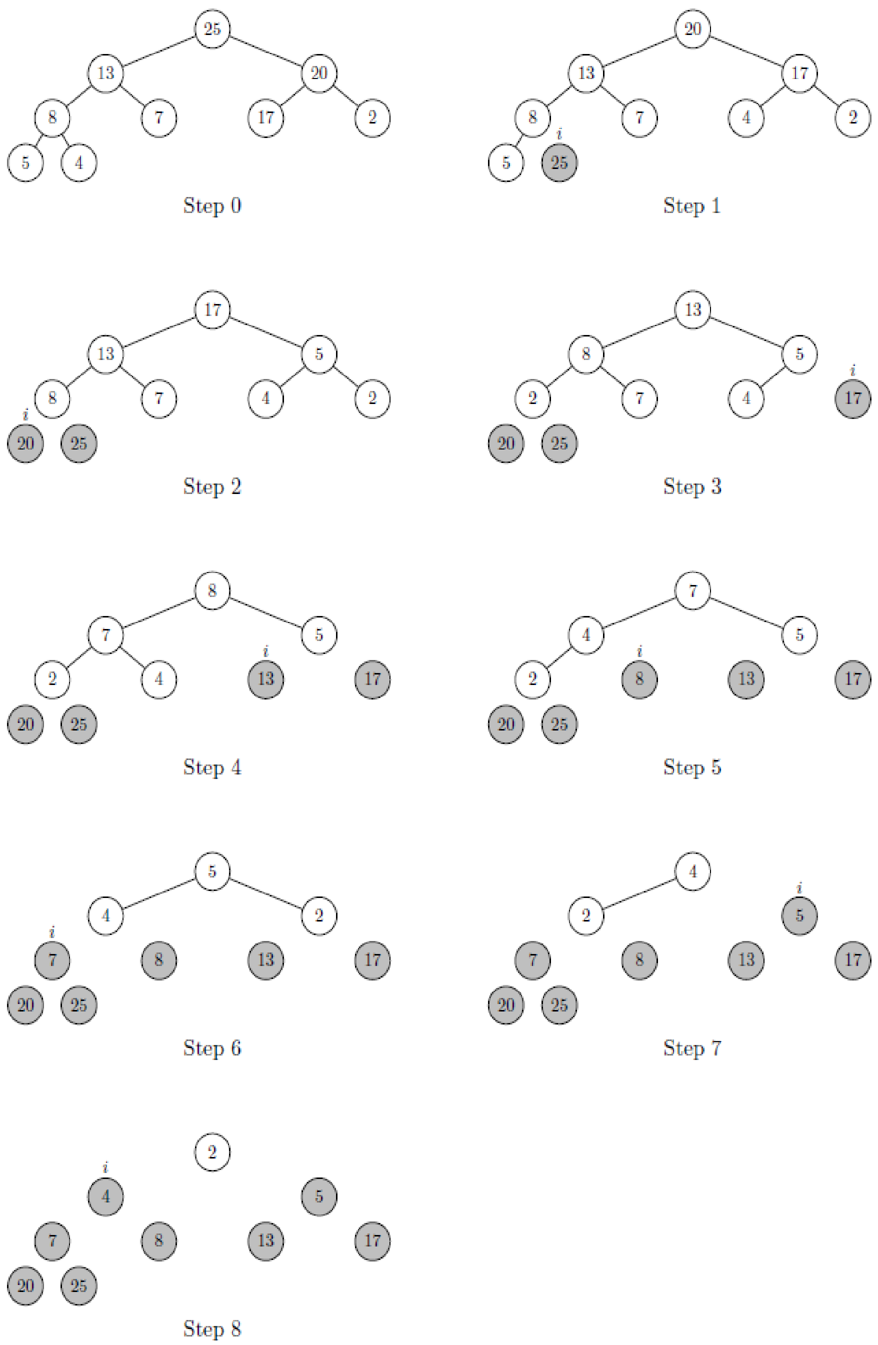


MAX-HEAPIFY$(A, 2)$:



MAX-HEAPIFY$(A, 1)$:





This gives a final sorted array $A = \langle 2, 4, 5, 7, 8, 13, 17, 20, 25 \rangle$. ∎

7. A priority queue can be implemented as a heap because
   a. The root can be easily be identified as the topmost priority.
   b. The heap is not always sorted so any value can be the top priority.
   c. The heap always has a left bottom node that can be the top priority.
   d. None of the above.
   solution) a.