3.1 Assignment 1 – Heap

1/

Maximum numbers of elements is $2^h - 1$, happened when the lowest level is full.

Minimum numbers of elements is 2^{h-1}, happened when the lowest level only contain a single node.

2/

The smallest element in a max-heap might reside in the leaf nodes which stands at the end of the tree and do not have any children.

3/

Yes, because sorted array will have a[i] < a[j], the smallest value will occur first.

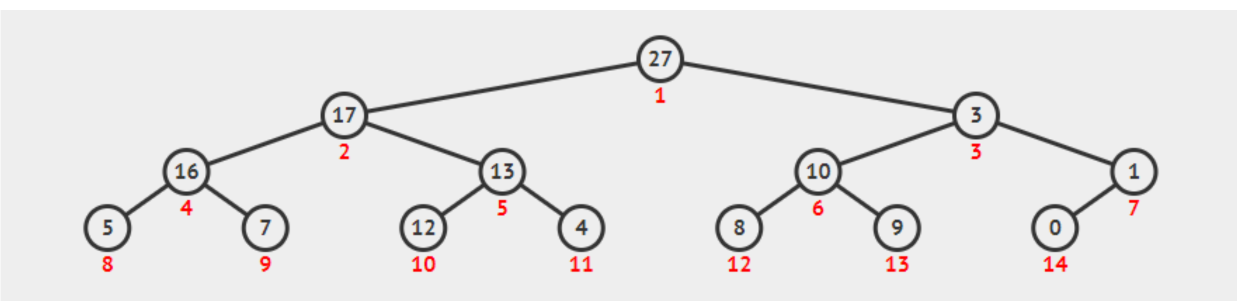
On the other hand, the min-heap requires the value of root to be the smallest, also it needs A[i] <= A[left[i]] and A[i]<=[right[i]], where i < left[i] and I < right[i].

4/

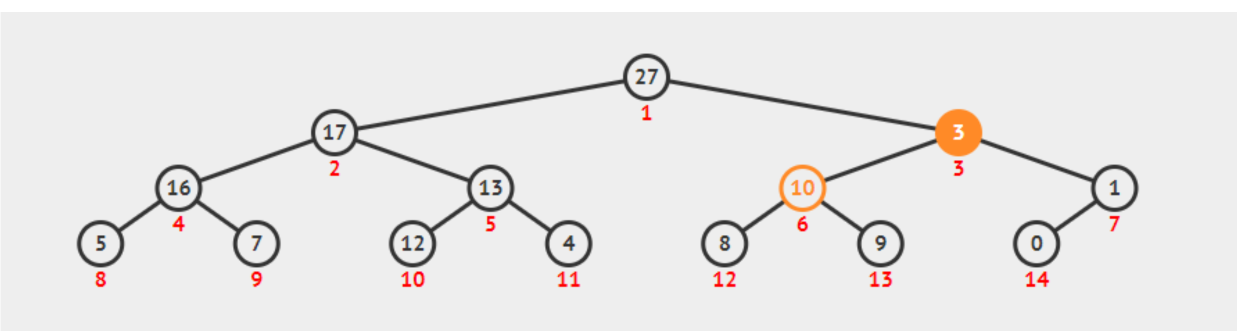
This array is not a max-heap because we have to rotate once when we insert 7. We have to switch 6 and 7.

5/

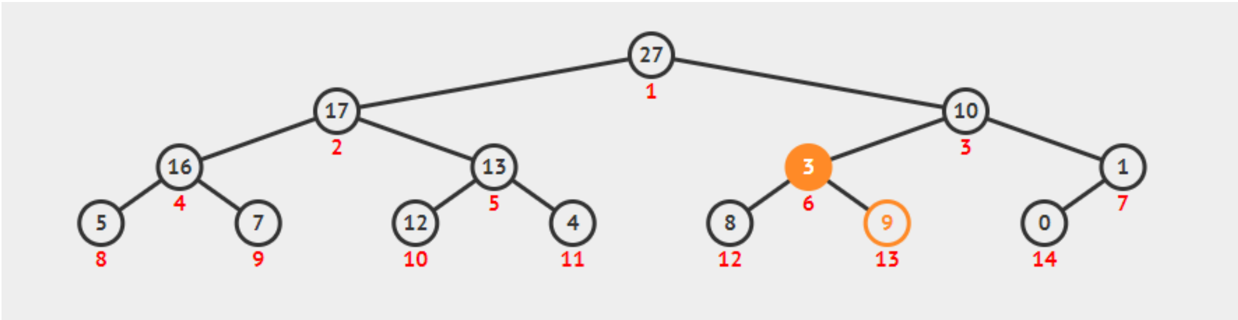
Step 1:



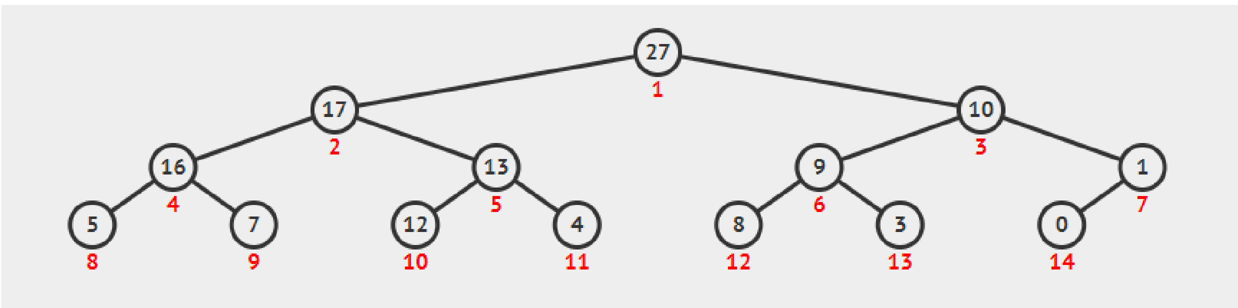
Step 2:



Step 3:



Step 4:



6/

Nothing will happen because A[i] has already larger than its children.

7/

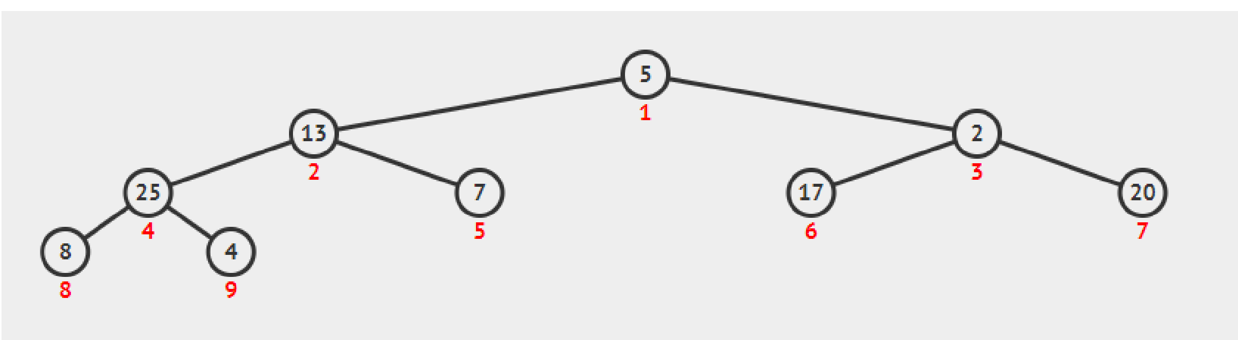
In this case, node i we called has no children, also left[i] and right [i] are larger than A.heapsize; therefore the array index is out of range, algorithm return errors.

8/

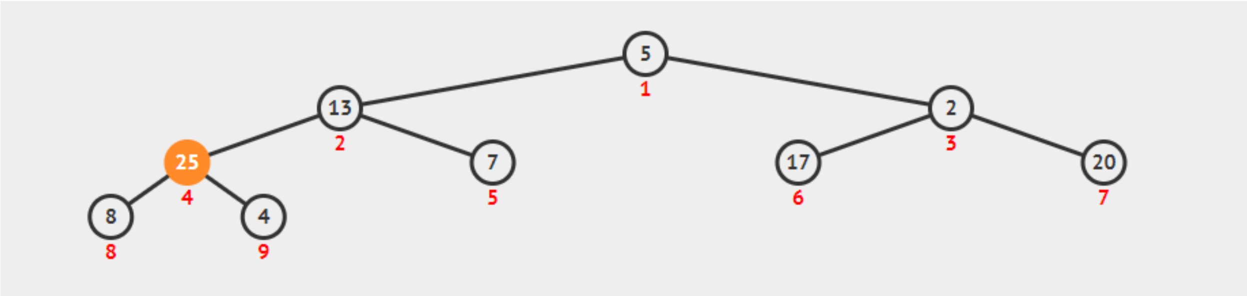
BUILD-MAX-HEAP

We have A.length = 9, so MAX-HEAPIFY (A,i), with i = 4, 3, 2, 1.

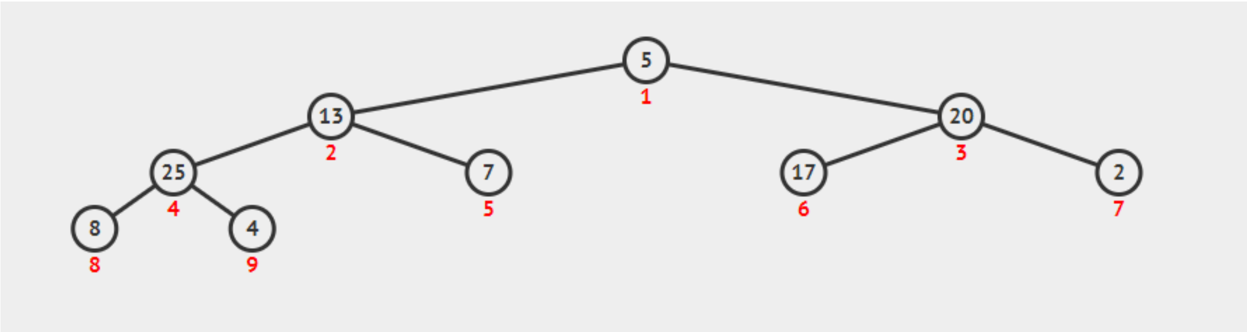
BUILD-MAX-HEAP (A)



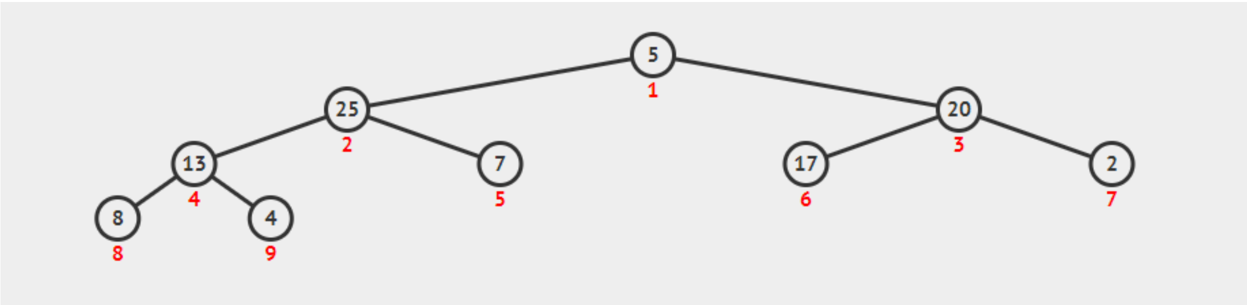
MAX-HEAPIFY (A,4)



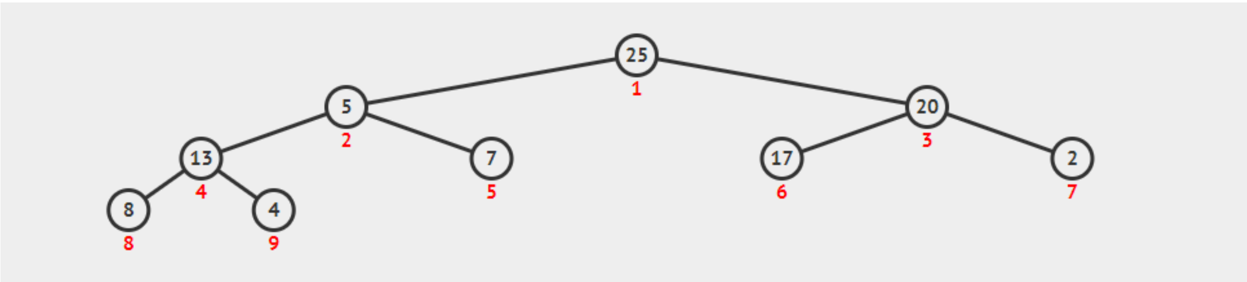
MAX-HEAPIFY (A,3)



MAX-HEAPIFY (A,2)



MAX-HEAPIFY (A,1)



10/

HEAPSORT

Step 1: Root stores the max value, take it out.

Step 2: Replace the root with the last leaf.

Step 3: Compare the value with the nodes below and swap until find out the largest number to stay on the root.

Repeat these steps until heap only has 1 node (root).

3.2 Assignment – Red black tree

1/

Red black trees: number 2 and number 5.

Not Red black tree:

- Tree 1: property 4 is violated.
- Tree 3: property 4 is violated.
- Tree 4: property 3 is violated.

2/

Black heights of all nodes is 2.

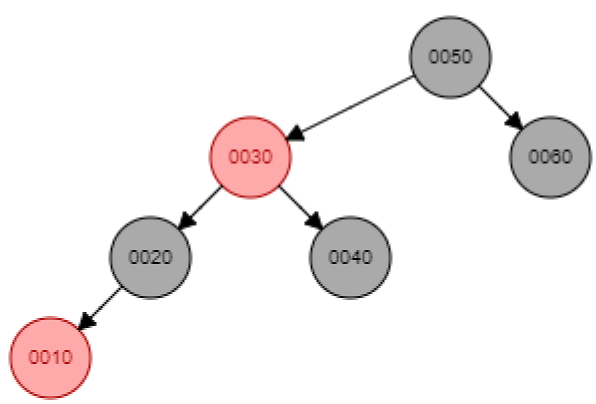
3/

It is possible to have red black tree contains all black nodes if this tree is perfectly balance (all leaves are at the same level, every parent has 2 children).

Example: Tree 2 in Ex1.

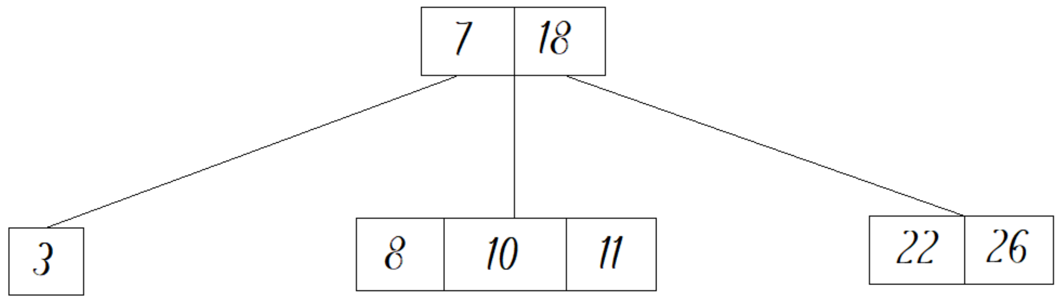
4/

Red black tree that is not AVL tree



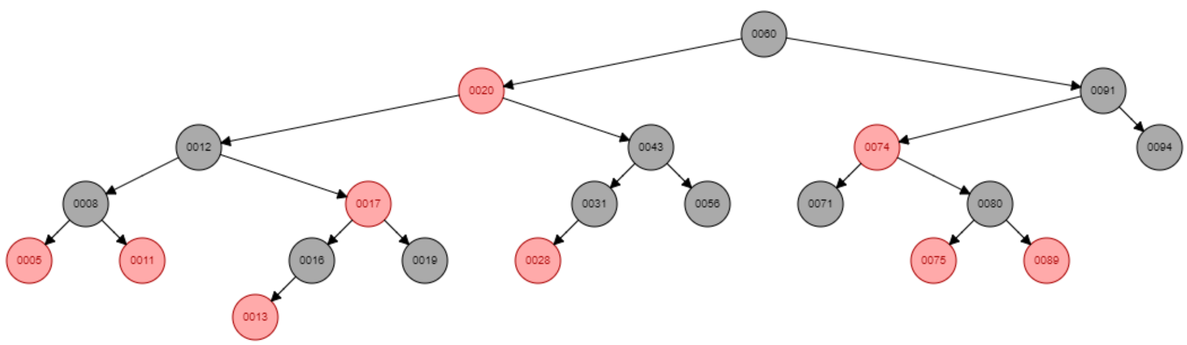
5/

Merge red nodes into black parents will get 2-3-4 tree.



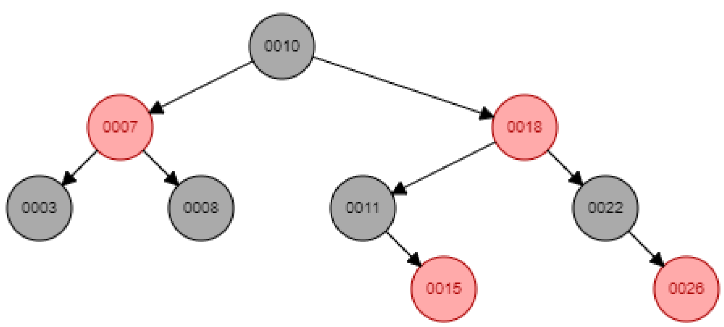
6/

Conver 2-3-4 tree into red black tree



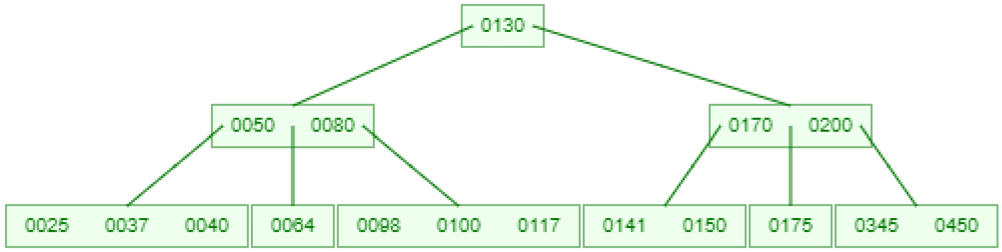
7/

Insert 15 into red black tree



8/

2-3-4 tree:



Red black tree: