Heap

A heap is a complete binary tree, where each node have value greater than or equal to all its descendants. That heap is called a Max Heap.

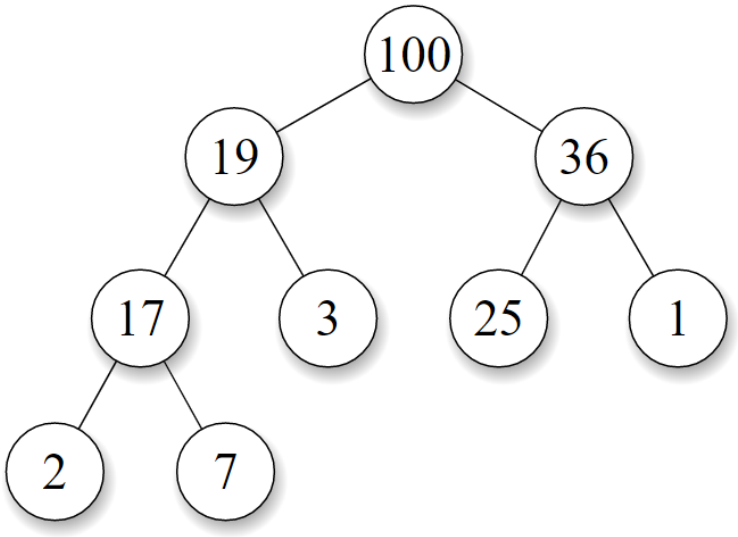


Figure : Max Heap

Note : Root node will have the largest value in a Max Heap

Min Heap

If in a heap, each node have value less than or equal to all its descendants, that heap is called Min Heap.

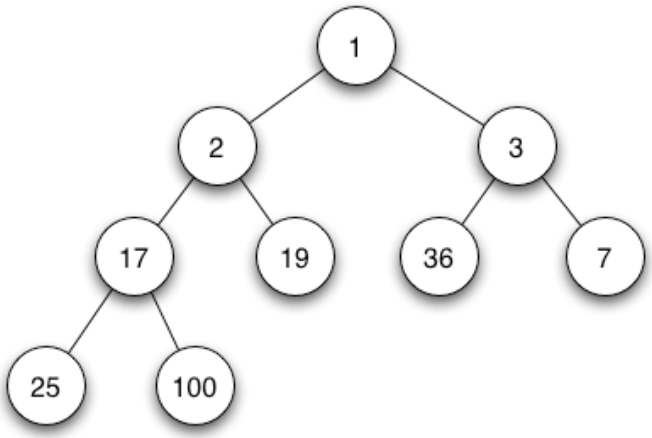


Figure : Min Heap

Note : Root node will have the smallest value in a Min Heap

Inserting a node in a Heap

Note : New node is always added at the last, and then moved upwards if its greater than its parent.

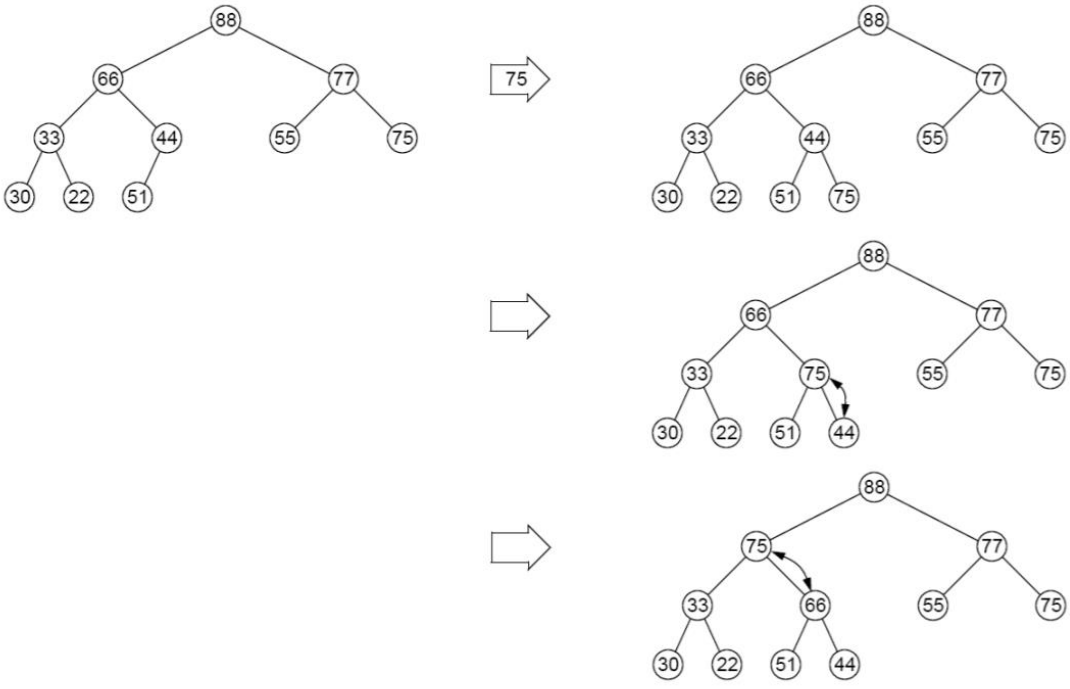


Figure : Inserting 75 into a Heap

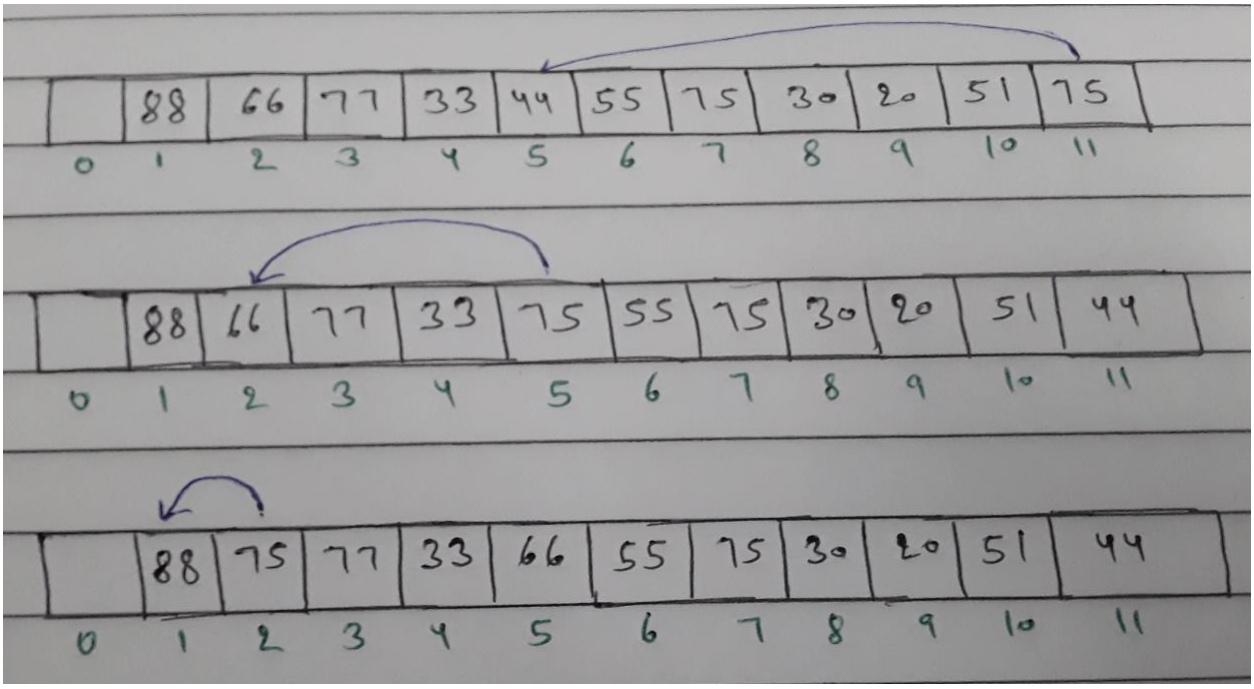
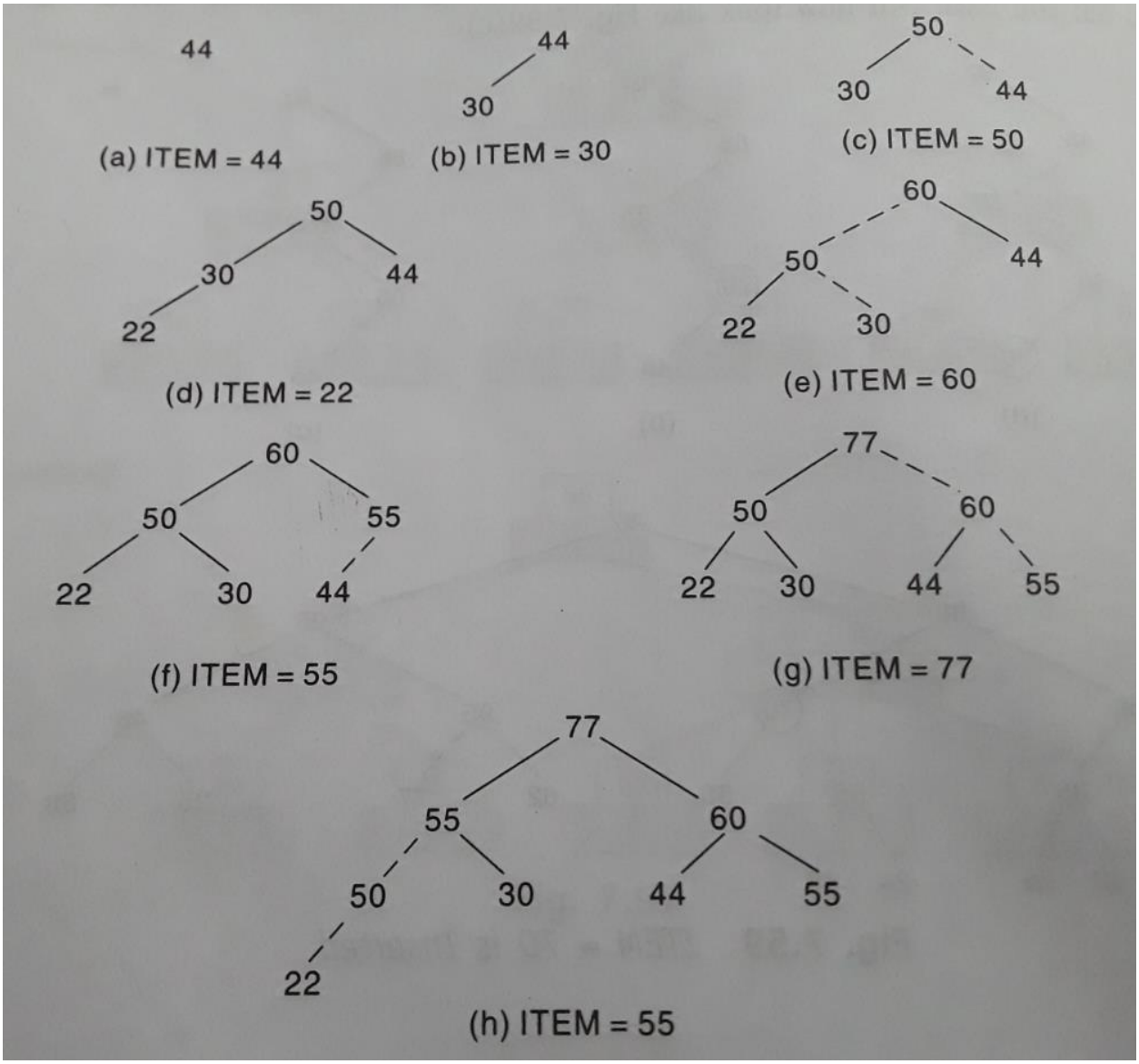


Figure : Inserting 75 into a Heap

Creating a Heap

[44, 30, 50, 22, 60, 55, 77, 55]



Deleting a node from Heap

Note : While deleting a node always root node of the heap is deleted and last node in the heap becomes the new root node, then it's compared with its children and if it is smaller than any of its children it is moved downwards.

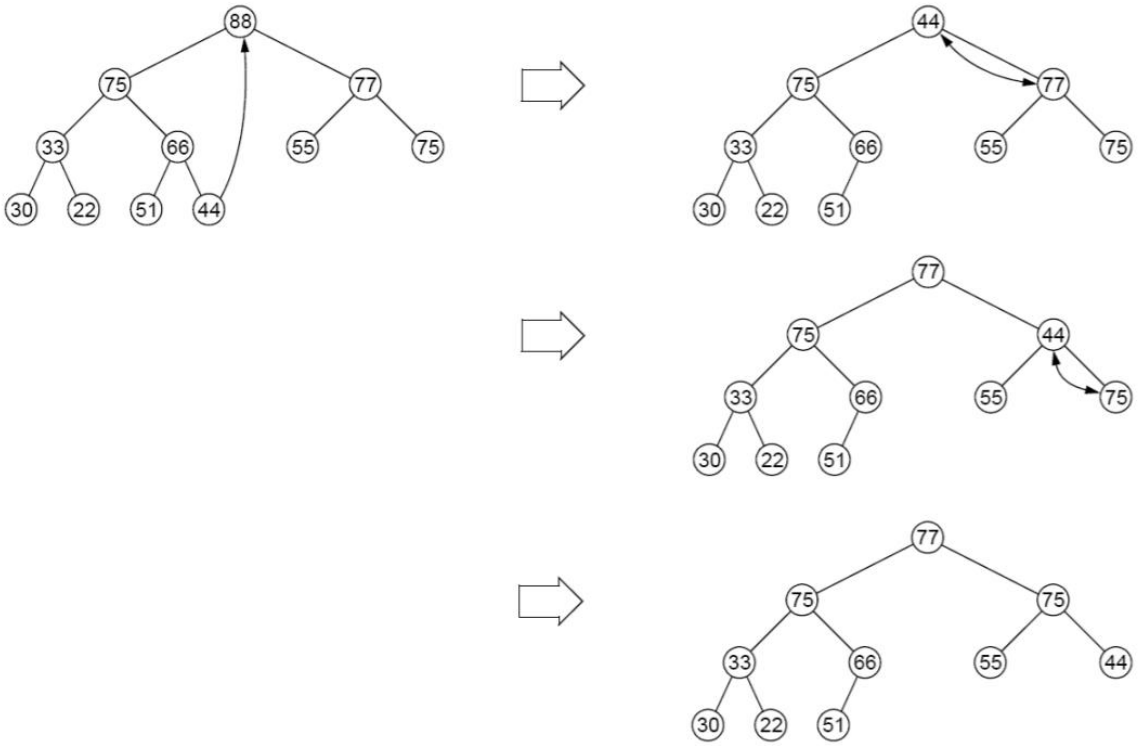


Figure : Deleting a node from Heap

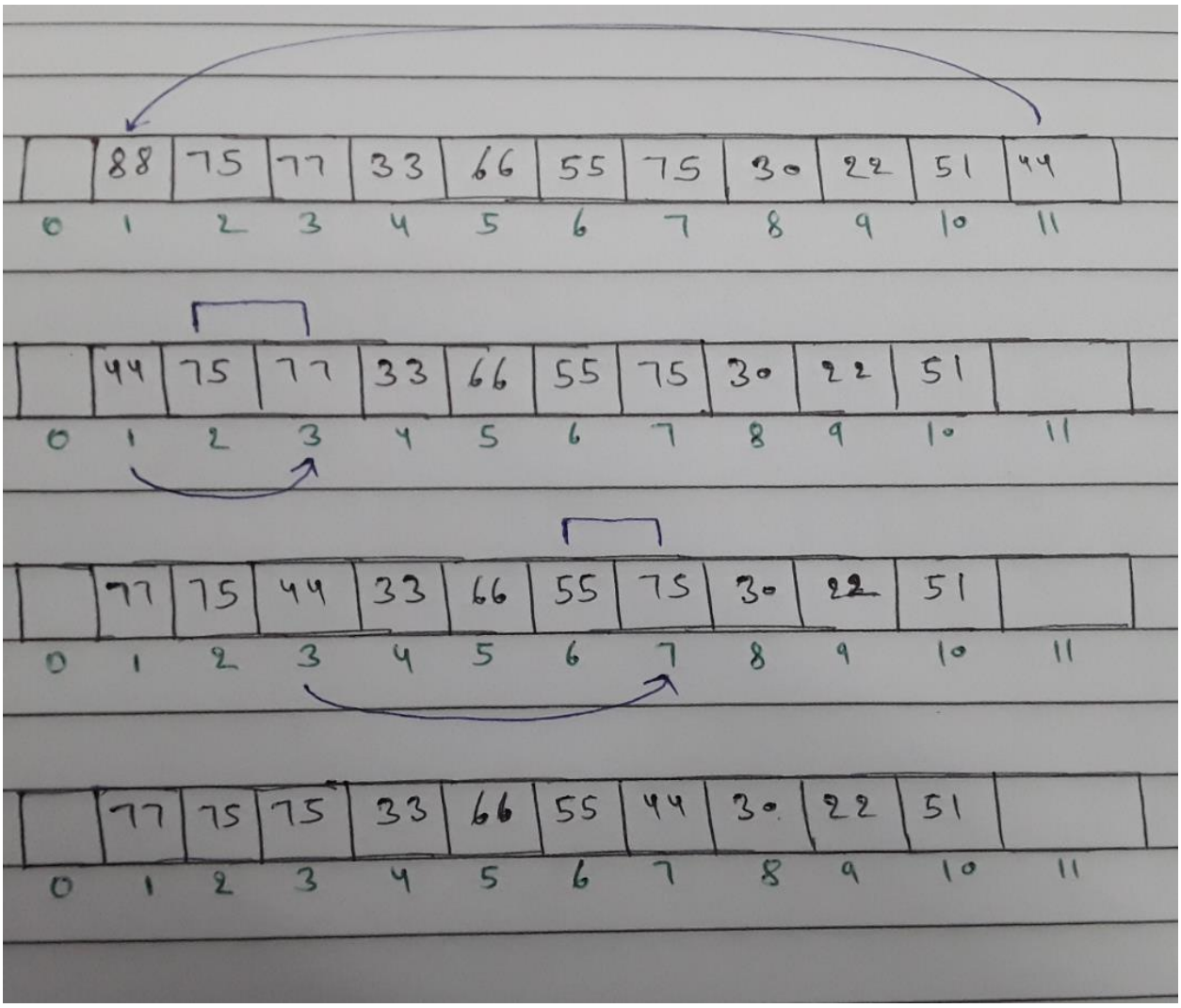
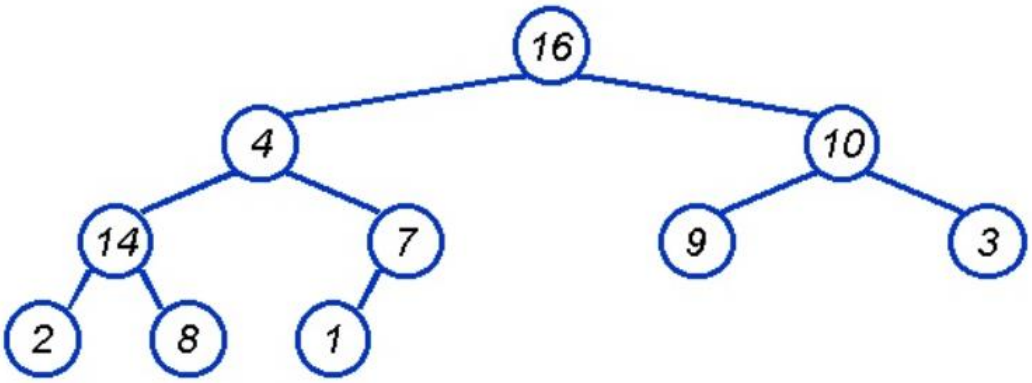
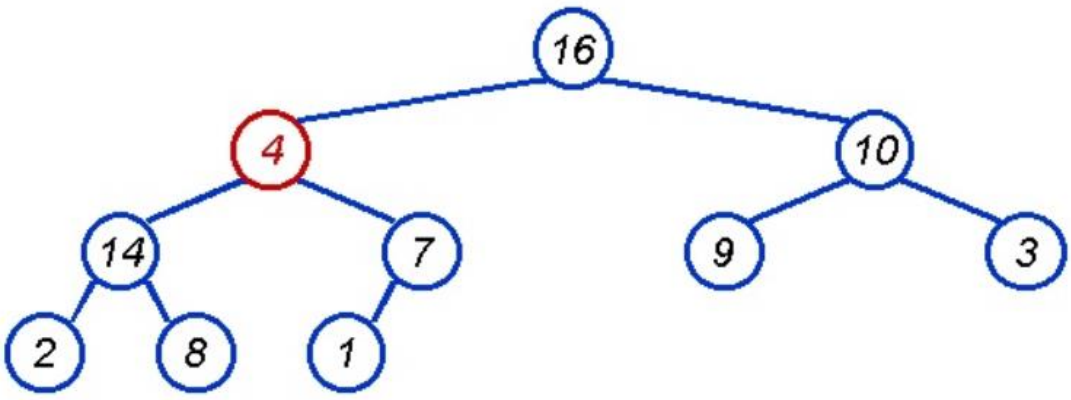


Figure : Deleting a node from Heap

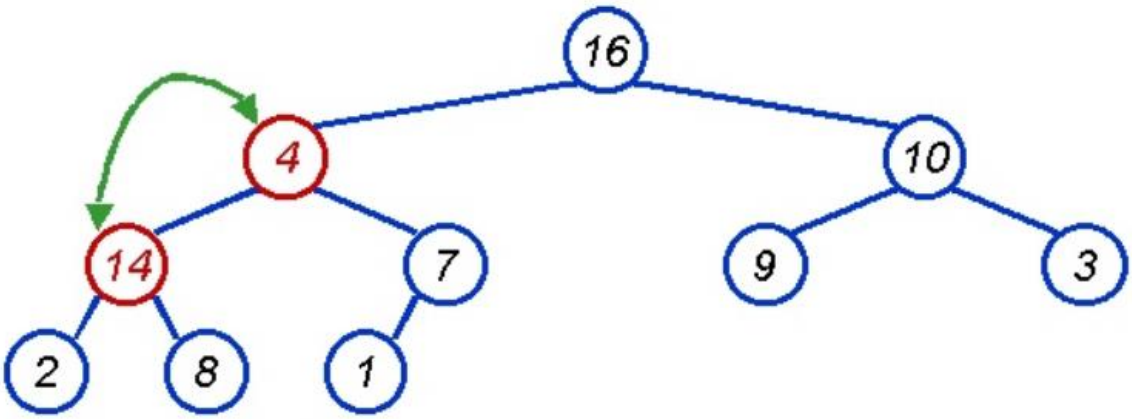
Heapify

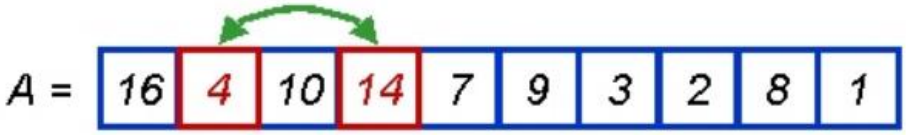

16
4
10
14
7
9
3
2
8
1

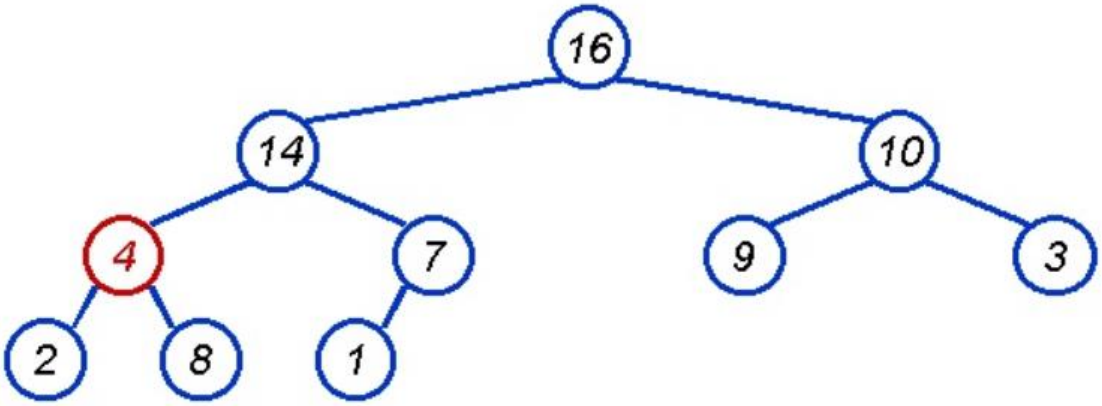

A =
16
4
10
14
7
9
3
2
8
1

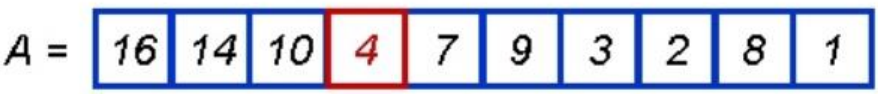

16
4
10
14
7
9
3
2
8
1

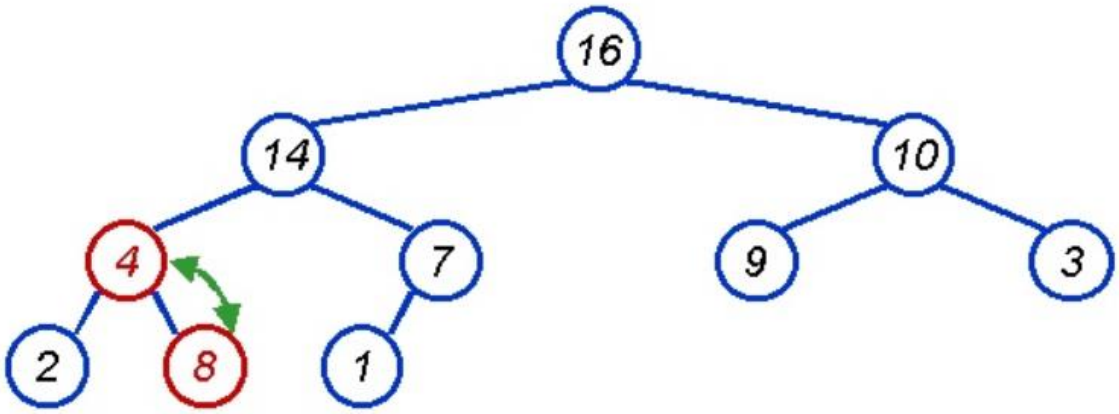

A =
16
4
10
14
7
9
3
2
8
1

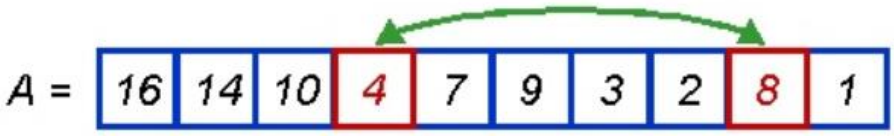

16
4
10
14
7
9
3
2
8
1

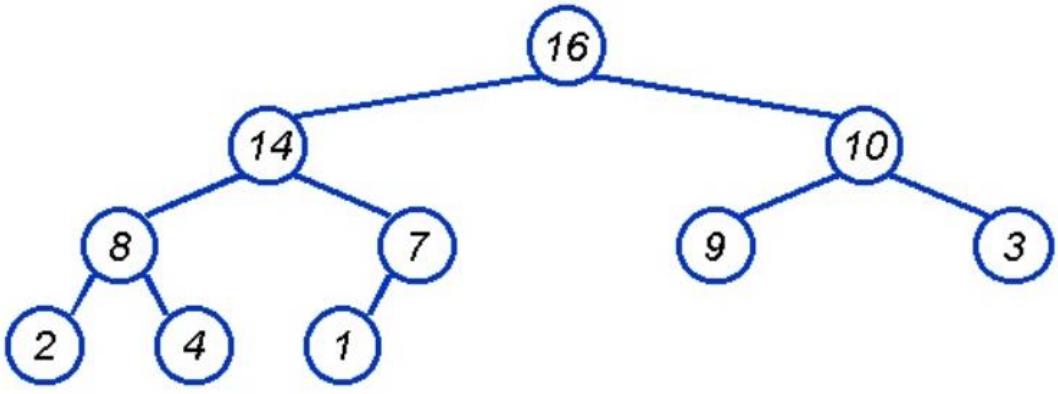

A =
16
4
10
14
7
9
3
2
8
1

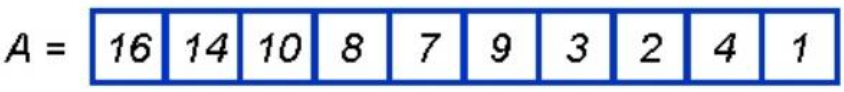

16
14
10
4
7
9
3
2
8
1


A =
16
14
10
4
7
9
3
2
8
1


16
14
10
4
7
9
3
2
8
1


A =
16
14
10
4
7
9
3
2
8
1


16
14
10
8
7
9
3
2
4
1


A =
16
14
10
8
7
9
3
2
4
1