Priority Queues – 1

1. **Insertion In Max Heap**

**Send Feedback**

**Consider a max heap, represented by the array: 40, 30, 20, 10, 15, 16, 17, 8, 4. Now consider that a value 35 is inserted into this heap. After insertion, the new heap is**

1. 40, 30, 20, 10, 15, 16, 17, 8, 4, 35
2. 40, 35, 20, 10, 30, 16, 17, 8, 4, 15  answer
3. 40, 30, 20, 10, 35, 16, 17, 8, 4, 15
4. 40, 35, 20, 10, 15, 16, 17, 8, 4, 30
5. **Insertion In PQ-MaxHeap**

**Send Feedback**

#### **A Priority-Queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is given below: 10, 8, 5, 3, 2 Two new elements ”1‘ and ”7‘ are inserted in the heap in that order. The level-order traversal of the heap after the insertion of the elements is:**

1. 10, 8, 7, 5, 3, 2, 1
2. 10, 8, 7, 2, 3, 1, 5
3. 10, 8, 7, 1, 2, 3, 5
4. 10, 8, 7, 3, 2, 1, 5  answer
5. **Re-heap After Removal**

**Send Feedback**

**An operator delete(i) for a binary heap data structure is to be designed to delete the item in the i-th node. Assume that the heap is implemented in an array and i refers to the i-th index of the array. If the heap tree has depth d (number of edges on the path from the root to the farthest leaf), then what is the time complexity to re-fix the heap efficiently after the removal of the element?**

1. O(1)
2. O(d) but not O(1)  answer
3. O(2^d ) but not O(d)
4. O(d\*(2^d) ) but not O(2^d )
5. **Code : Remove Min**

**Send Feedback**

#### **Implement the function RemoveMin for the min priority queue class.**

#### **For a minimum priority queue, write the function for removing the minimum element present. Remove and return the minimum element.**

##### Note : main function is given for your reference which we are using internally to test the code.

1. **Code : Max Priority Queue**

**Send Feedback**

#### **Implement the class for Max Priority Queue which includes following functions -**

##### 1. getSize -

#### **Return the size of priority queue i.e. number of elements present in the priority queue.**

##### 2. isEmpty -

#### **Check if priority queue is empty or not. Return true or false accordingly.**

##### 3. insert -

#### **Given an element, insert that element in the priority queue at the correct position.**

##### 4. getMax -

#### **Return the maximum element present in the priority queue without deleting. Return -Infinity if priority queue is empty.**

##### 5. removeMax -

#### **Delete and return the maximum element present in the priority queue. Return -Infinity if priority queue is empty.**

##### Note : main function is given for your reference which we are using internally to test the class.