Priority Queues – 2

1. **Inplace Heap Sort**

**Send Feedback**

Given an integer array of size n. Sort this array (in decreasing order) using heap sort.

Space complexity should be O(1).

**Input Format :**

Line 1 : Integer n, Array size

Line 2 : Array elements, separated by space

**Output Format :**

Array elements after sorting

**Constraints :**

1 <= n <= 10^6

**Sample Input:**

6

2 6 8 5 4 3

**Sample Output:**

8 6 5 4 3 2

1. **K Smallest Elements**

**Send Feedback**

You are given with an integer k and an array of integers that contain numbers in random order. Write a program to find k smallest numbers from given array. You need to save them in an array and return it.

Time complexity should be O(nlogk) and space complexity should be not more than O(k).

**Order of elements in the output is not important.**

**Input Format :**

Line 1 : Size of array (n)

Line 2 : Array elements (separated by space)

Line 3 : Integer k

**Output Format :**

k smallest elements

**Sample Input 1 :**

13

2 12 9 16 10 5 3 20 25 11 1 8 6

4

**Sample Output 1 :**

5

3

2

1

1. **K Largest Elements**

**Send Feedback**

You are given with an integer k and an array of integers that contain numbers in random order. Write a program to find k largest numbers from given array. You need to save them in an array and return it.

Time complexity should be O(nlogk) and space complexity should be not more than O(k).

**Order of elements in the output is not important.**

**Input Format :**

Line 1 : Size of array (n)

Line 2 : Array elements (separated by space)

Line 3 : Integer k

**Output Format :**

k largest elements

**Sample Input :**

13

2 12 9 16 10 5 3 20 25 11 1 8 6

4

**Sample Output :**

12

16

20

25

Assignment

1. **Check Max-Heap**

**Send Feedback**

Given an array of integers, check whether it represents max-heap or not.

Return true or false.

**Input Format :**

Line 1 : An integer N i.e. size of the array

Line 2 : N integers which are elements of the array, separated by spaces

**Output Format :**

true if it represents max-heap and false if it is not a max-heap

**Constraints :**

1 <= N <= 10^5

1 <= Ai <= 10^5

**Sample Input :**

8

42 20 18 6 14 11 9 4

**Sample Output :**

true

1. **Kth largest element**

**Send Feedback**

Given an array A of random integers and an integer k, find and return the kth largest element in the array.

Try to do this question in less than O(nlogn) time.

**Input Format :**

Line 1 : An integer N i.e. size of the array

Line 2 : N integers which are elements of the array, separated by spaces

Line 3 : An integer k

**Output Format :**

kth largest element

**Input Constraints :**

1 <= N, Ai, k <= 10^5

**Sample Input 1 :**

6

9 4 8 7 11 3

2

**Sample Output 1 :**

9

**Sample Input 2 :**

8

2 6 10 11 13 4 1 20

4

**Sample Output 2 :**

10

1. **Buy the ticket**

**Send Feedback**

You want to buy a ticket for a well-known concert which is happening in your city. But the number of tickets available is limited. Hence the sponsors of the concert decided to sell tickets to customers based on some priority.

A queue is maintained for buying the tickets and every person has attached with a priority (an integer, 1 being the lowest priority). The tickets are sold in the following manner -

1. The first person (pi) in the queue asked to comes out.

2. If there is another person present in the queue who has higher priority than pi, then ask pi to move at end of the queue without giving him the ticket.

3. Otherwise, give him the ticket (and don't make him stand in queue again).

**Giving a ticket to a person takes exactly 1 minutes and it takes no time for removing and adding a person to the queue. And you can assume that no new person joins the queue.**

Given a list of priorities of N persons standing in the queue and the index of your priority (indexing starts from 0). Find and return the time it will take until you get the ticket.

Input Format :

Line 1 : Integer N (Total number of people standing in queue)

Line 2 : Priorities of every person (n space separated integers)

Line 3 : Integer k (index of your priority)

**Output Format :**

Time required

**Sample Input 1 :**

3

3 9 4

2

**Sample Output 1 :**

2

**Sample Output 1 Explanation :**

Person with priority 3 comes out. But there is a person with higher priority than him. So he goes and then stands in the queue at the end. Queue's status : {9, 4, 3}. Time : 0 secs.

Next, the person with priority 9 comes out. And there is no person with higher priority than him. So he'll get the ticket. Queue's status : {4, 3}. Time : 1 secs.

Next, the person with priority 4 comes out (which is you). And there is no person with higher priority than you. So you'll get the ticket. Time : 2 secs.

**Sample Input 2 :**

5

2 3 2 2 4

3

**Sample Output 2 :**

4

Assignment