



Dash



All



Articles



Videos



Problems



Quiz

&lt;&lt; Prev

Next &gt;&gt;

## Sort K-Sorted Array

We can sort such arrays **efficiently with the help of Heap data structure**. Following is the detailed process that uses Heap.

- 1) Create a Min Heap of size  $k+1$  with first  $k+1$  elements. This will take  $O(k)$  time (See this GFact). We are creating heap of size  $k$  as the element can be atmost  $k$  distance from its index in a sorted array.
- 2) One by one remove min element from heap, put it in result array, and add a new element to heap from remaining elements.

Removing an element and adding a new element to min heap will take  $\log k$  time. So overall complexity will be  $O(k) + O((n-k) * \log(k))$ .

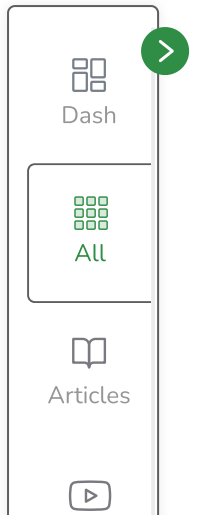
C++

Java

```
// A java program to sort a nearly sorted array
import java.util.Iterator;
import java.util.PriorityQueue;

class GFG {
    private static void kSort(int[] arr, int n, int k)
    {
        if (arr == null || arr.length == 0) {
            return;
        }
        // min heap
        PriorityQueue<Integer> priorityQueue
            = new PriorityQueue<>();
        // if there are less than k + 1 elements present in the array
        int minCount = Math.min(arr.length, k + 1);
```





Get 90% Refund!

Courses

Tutorials

Jobs

Practice

Contests



```
// add first k + 1 items to the min heap
for (int i = 0; i < minCount; i++) {
    priorityQueue.add(arr[i]);
}

int index = 0;
for (int i = k + 1; i < n; i++) {
    arr[index++] = priorityQueue.peek();
    priorityQueue.poll();
    priorityQueue.add(arr[i]);
}

Iterator<Integer> itr = priorityQueue.iterator();

while (itr.hasNext()) {
    arr[index++] = priorityQueue.peek();
    priorityQueue.poll();
}
```

```
private static void printArray(int[] arr, int n)
{
    for (int i = 0; i < n; i++)
        System.out.print(arr[i] + " ");
}

// Driver Code
public static void main(String[] args)
{
    int k = 3;
    int arr[] = { 2, 6, 3, 12, 56, 8 };
    int n = arr.length;
    kSort(arr, n, k);
    System.out.println("Following is sorted array");
    printArray(arr, n);
}
}
```



P

```
// This code is contributed by  
// Manpreet Singh(manpreetsngh294)
```

## Output

Following is sorted array

2 3 6 8 12 56

**Time Complexity:**  $O(k) + O((m) * \log(k))$ , where  $m = n - k$

**Auxiliary Space:**  $O(k)$

 Report An Issue

If you are facing any issue on this page. Please let us know.

Mark as Read



Dash



All



Articles



Videos



Problems



Quiz

