

# Kth Smallest Number (easy)

We'll cover the following ^

- Problem Statement
- Try it yourself
- Solution
- Code
  - Time complexity
  - Space complexity
- An Alternate Approach

## Problem Statement #

Given an unsorted array of numbers, find Kth smallest number in it.

Please note that it is the Kth smallest number in the sorted order, not the Kth distinct element.

Note: For a detailed discussion about different approaches to solve this problem, take a look at [Kth Smallest Number](https://www.educative.io/collection/page/5668639101419520/5671464854355968/4817079184130048)

(<https://www.educative.io/collection/page/5668639101419520/5671464854355968/4817079184130048>).

### Example 1:

Input: [1, 5, 12, 2, 11, 5], K = 3

Output: 5

Explanation: The 3rd smallest number is '5', as the first two smaller numbers are [1, 2].

### Example 2:

Input: [1, 5, 12, 2, 11, 5], K = 4

Output: 5

Explanation: The 4th smallest number is '5', as the first three small numbers are [1, 2, 5].

### Example 3:

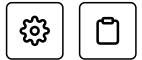
Input: [5, 12, 11, -1, 12], K = 3

Output: 11

Explanation: The 3rd smallest number is '11', as the first two small numbers are [5, -1].

## Try it yourself #

Try solving this question here:



Java	Python3	JS	C++
------	---------	----	-----

```
1 def find_Kth_smallest_number(nums, k):
2     # TODO: Write your code here
3     return -1
4
5
6 def main():
7
8     print("Kth smallest number is: " +
9           str(find_Kth_smallest_number([1, 5, 12, 2, 11, 5], 3)))
10
11     # since there are two 5s in the input array, our 3rd and 4th smallest numbers should be
12     print("Kth smallest number is: " +
13           str(find_Kth_smallest_number([1, 5, 12, 2, 11, 5], 4)))
14
15     print("Kth smallest number is: " +
16           str(find_Kth_smallest_number([5, 12, 11, -1, 12], 3)))
17
18
19 main()
20
```

## Solution #

This problem follows the Top 'K' Numbers

(<https://www.educative.io/collection/page/5668639101419520/5671464854355968/5728885882748928/>) pattern but has two differences:

1. Here we need to find the Kth smallest number, whereas in Top 'K' Numbers (<https://www.educative.io/collection/page/5668639101419520/5671464854355968/5728885882748928/>) we were dealing with 'K' largest numbers.
2. In this problem, we need to find only one number (Kth smallest) compared to finding all 'K' largest numbers.

We can follow the same approach as discussed in the 'Top K Elements' problem. To handle the first difference mentioned above, we can use a max-heap instead of a min-heap. As we know, the root is the biggest element in the max heap. So, since we want to keep track of the 'K' smallest numbers, we can compare every number with the root while iterating through all numbers, and if it is smaller than the root, we'll take the root out and insert the smaller number.

## Code #

Here is what our algorithm will look like:

Java	Python3	C++	JS
------	---------	-----	----

```
1 from heapq import *
2
3
```

```

4 def find_Kth_smallest_number(nums, k):
5     maxHeap = []
6     # put first k numbers in the max heap
7     for i in range(k):
8         heappush(maxHeap, -nums[i])
9
10    # go through the remaining numbers of the array, if the number from the array is smaller
11    # top(biggest) number of the heap, remove the top number from heap and add the number fr
12    for i in range(k, len(nums)):
13        if -nums[i] > maxHeap[0]:
14            heappop(maxHeap)
15            heappush(maxHeap, -nums[i])
16
17    # the root of the heap has the Kth smallest number
18    return -maxHeap[0]
19
20
21 def main():
22
23     print("Kth smallest number is: " +
24           str(find_Kth_smallest_number([1, 5, 12, 2, 11, 5], 3)))
25
26     # since there are two 5s in the input array, our 3rd and 4th smallest numbers should be
27     print("Kth smallest number is: " +
28           str(find_Kth_smallest_number([1, 5, 12, 2, 11, 5], 4)))

```



### Time complexity #

The time complexity of this algorithm is  $O(K * \log K + (N - K) * \log K)$ , which is asymptotically equal to  $O(N * \log K)$

### Space complexity #

The space complexity will be  $O(K)$  because we need to store 'K' smallest numbers in the heap.

## An Alternate Approach #

Alternatively, we can use a **Min Heap** to find the Kth smallest number. We can insert all the numbers in the min-heap and then extract the top 'K' numbers from the heap to find the Kth smallest number. Initializing the min-heap with all numbers will take  $O(N)$  and extracting 'K' numbers will take  $O(K \log N)$ . Overall, the time complexity of this algorithm will be  $O(N + K \log N)$  and the space complexity will be  $O(N)$ .

← Back

Next →

Top 'K' Numbers (easy)

'K' Closest Points to the Origin (easy)

✓ Mark as Completed



Report an  
Issue



Ask a Question

([https://discuss.educative.io/tag/kth-smallest-number-easy\\_\\_pattern-top-k-elements\\_\\_grokking-the-coding-interview-patterns-for-coding-questions](https://discuss.educative.io/tag/kth-smallest-number-easy__pattern-top-k-elements__grokking-the-coding-interview-patterns-for-coding-questions))

