K’th Smallest/Largest Element in Unsorted Array | Set 1

Given an array and a number k where k is smaller than size of array, we need to find the k’th smallest element in the given array. It is given that ll array elements are distinct.

**Examples:**

***Input****: arr[] = {7, 10, 4, 3, 20, 15}  
k = 3****Output****: 7*

***Input****: arr[] = {7, 10, 4, 3, 20, 15}  
k = 4****Output****: 10*

**Method 1 (Simple Solution)**

A simple solution is to sort the given array using a O(N log N) sorting algorithm like [Merge Sort](http://geeksquiz.com/merge-sort/), [Heap Sort](http://geeksquiz.com/heap-sort/), etc and return the element at index k-1 in the sorted array.

Time Complexity of this solution is O(N Log N)

# Python3 program to find k'th smallest

# element

# Function to return k'th smallest

# element in a given array

def kthSmallest(arr, n, k):

    # Sort the given array

    arr.sort()

    # Return k'th element in the

    # sorted array

    return arr[k-1]

# Driver code

if \_\_name\_\_=='\_\_main\_\_':

    arr = [12, 3, 5, 7, 19]

    n = len(arr)

    k = 2

    print("K'th smallest element is",

          kthSmallest(arr, n, k))

Another Approach:-

class Solution(object):

def findKthLargest(self, nums, k):

i = 1

while i < k:

a = max(nums)

nums.remove(a)

i+=1

return max(nums)

class Solution:

def findKthLargest(self, arr: List[int], k: int) -> int:

if k>len(arr) or not arr or k<0: return

import heapq

heapq.heapify(arr)

for i in range(len(arr)-k+1):

pop = heapq.heappop(arr)

return pop

# Find the k most frequent words from data set in Python

from collections import Counter

word\_set = " This is a series of strings to count " \

   "many words . They sometime hurt and words sometime inspire "\

   "Also sometime fewer words convey more meaning than a bag of words "\

   "Be careful what you speak or what you write or even what you think of. "\

# Create list of all the words in the string

word\_list = word\_set.split()

# Get the count of each word.

word\_count = Counter(word\_list)

# Use most\_common() method from Counter subclass

print(word\_count.most\_common(3))

## Output

Running the above code gives us the following result −

[('words', 4), ('sometime', 3), ('what', 3)]