

## Solution Review: Find k Largest Elements in the List

We'll cover the following





- Solution #1: Creating a Max-Heap and removing max  $k$  times
  - Time Complexity
- Solution #2: Using Quickselect
  - Time Complexity

### Solution #1: Creating a Max-Heap and removing max $k$ times #

main.py

MaxHeap.py

```
1 from MaxHeap import MaxHeap
2
3
4 def findKLargest(lst, k):
5     heap = MaxHeap() # Create a MaxHeap
6     # Populate the MaxHeap with elements of lst
7     heap.buildHeap(lst)
8     # Create a list such that:
9     # It has k elements where
10    # the k elements are the first k
11    # elements received from calling removeMax()
12    kLargest = [heap.removeMax() for i in range(k)]
13    return kLargest
14
15
16 lst = [9, 4, 7, 1, -2, 6, 5]
17 k = 3
18 print(findKLargest(lst, k))
19
```



We first create a max-heap out of the given list by inserting the list elements into an empty heap on **line 7**. We then call `removeMax()` on the heap  $k$  times, save the output in a list, and return it.

#### Time Complexity #

The time complexity of creating a heap is  $O(n)$  and removing min is  $O(k \log n)$ . So the total time complexity is  $O(n + k \log n)$  which is the same as  $O(k \log n)$ .

### Solution #2: Using Quickselect #

You can optimize this further by calling the Quick Select

(<https://en.wikipedia.org/wiki/Quickselect>) algorithm on the given list  $k$  times where the input to the algorithm goes from  $n$  till  $n - k$ . We have not presented the code here because it is not relevant to heaps, but we felt that the optimal solution should be mentioned.

## Time Complexity #

The *average-case* complexity of quick select is  $O(n)$ . So when called  $k$  times it will be in  $O(nk) \rightarrow O(n)$ .

**Note:** By looking at this problem, we can see how the heap can be used to solve the Find Second Maximum Value in a List (<https://www.educative.io/courses/data-structures-in-python-an-interview-refresher/YQrnIj3kx80>). All we have to do is set  $k$  to 2 and pick the second value!

← Back

Next →

Challenge 3: Find k largest elements in...

Heap Quiz: Test your understanding of...

✓ Mark as Completed



Report an  
Issue



Ask a Question

([https://discuss.educative.io/tag/solution-review-find-k-largest-elements-in-the-list\\_\\_introduction-to-heap\\_\\_data-structures-for-coding-interviews-in-python](https://discuss.educative.io/tag/solution-review-find-k-largest-elements-in-the-list__introduction-to-heap__data-structures-for-coding-interviews-in-python))