

Time Complexity

The alternate method to using heaps is to sort the array and then pick the Kth largest element. This sorting based method would take $O(N \log N)$ time to sort and $O(1)$ to lookup the Kth element, overall taking $O(N \log N)$ time.

The heap based method improves on this with a solution that works in $O(N \log K)$ time. The complexity of inserting elements into a heap that is K element large is $O(\log K)$ and we do this for N elements, giving us $O(N \log K)$. When K is small compared to N, this will provide significant improvement.

Commonly asked questions

- Find first K smallest/largest elements in the array
- Find the Kth largest/smallest element in an array
- Find the K closest points to the origin
- Find the top K frequent elements/words
- Frequency sort (sort a string based on frequency of characters)
- Remove K numbers from array to have maximum distinct numbers left
- Merge K sorted lists
- Find the continuous median of an array
- Reorganize a string so that no two adjacent characters are the same
- Find if a given array represents a Binary Max Heap
- Connect N ropes with minimum cost
- Convert BST to Min Heap
- Find the Kth smallest element in Sorted matrix
- Scheduling tasks (Hard)
- Minimum Meeting rooms (Hard)
- Skyline problem (Hard)
- Employee free time (Hard)