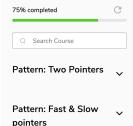


# Grokking the Coding Interview: Patterns for Coding Questions



## Pattern: Merge Intervals



## Pattern: In-place Reversal of a LinkedList





















Top 'K' Frequent Numbers (medium)

Frequency Sort (medium)

Kth Largest Number in a Stream

(medium)
'K' Closest Numbers (medium)

Maximum Distinct Elements (medium)

Sum of Elements (medium)
Rearrange String (hard)

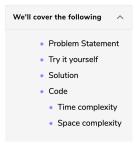
Problem Challenge 1
Solution Review: Problem
Challenge 1

Problem Challenge 2

Solution Review: Problem Challenge 2 Problem Challenge 3

Solution Review: Problem Challenge 3

## 'K' Closest Points to the Origin (easy)



## Problem Statement

Given an array of points in the a 2D plane, find 'K' closest points to the origin.

#### Example 1:

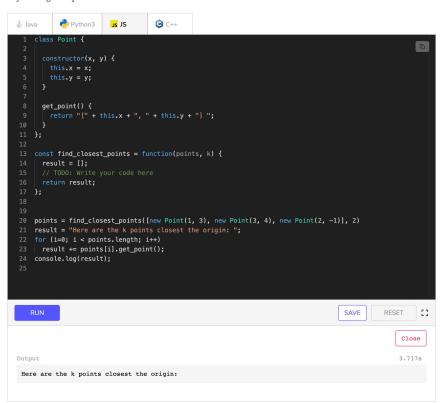
```
Input: points = [[1,2],[1,3]], K = 1
Output: [[1,2]]
Explanation: The Euclidean distance between (1, 2) and the origin is sqrt(5).
The Euclidean distance between (1, 3) and the origin is sqrt(10).
Since sqrt(5) < sqrt(10), therefore (1, 2) is closer to the origin.</pre>
```

## Example 2:

```
Input: point = [[1, 3], [3, 4], [2, -1]], K = 2
Output: [[1, 3], [2, -1]]
```

### Try it yourself

Try solving this question here:



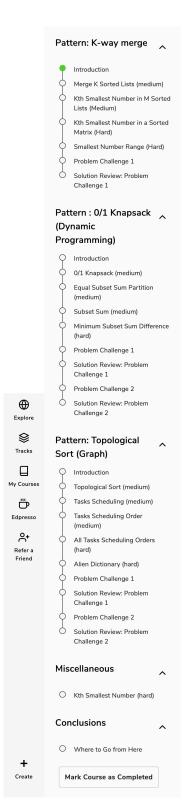
### Solution

The Euclidean distance of a point P(x,y) from the origin can be calculated through the following formula:

$$\sqrt{x^2+y^2}$$

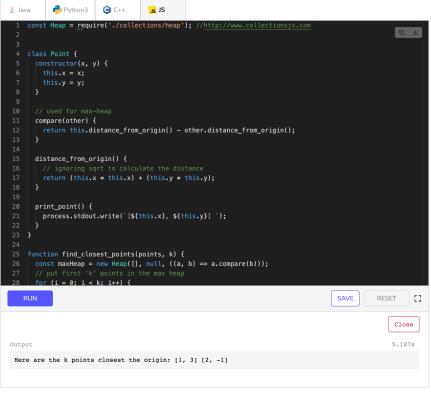
This problem follows the Top 'K' Numbers pattern. The only difference in this problem is that we need to find the closest point (to the origin) as compared to finding the largest numbers.

Following a similar approach, we can use a **Max Heap** to find 'K' points closest to the origin. While iterating through all points, if a point (say 'P') is closer to the origin than the top point of the max-heap, we will remove that top point from the heap and add 'P' to always keep the closest points in the heap.



### Code

Here is what our algorithm will look like:



Time complexity

The time complexity of this algorithm is (N\*logK) as we iterating all points and pushing them into the heap.

Space complexity

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

Interviewing soon? We've partnered with Hired so that companies apply to you instead of you  $\quad\times$  applying to them. See how  $\odot$ 

