

## sorting\_Searching

제로 이동 (Move Zeros)

k번째 제일큰 원소 (Kth Largest Element In An Array)

원점에 가장 가까운 지점 (K Closest Points to Origin)

미팅룸 (Meeting Room)

미팅룸 2 (Meeting Room2)

interval 병합 (Merge Interval)

로그 파일의 데이터 재정렬

- 1) **Arrays.sort()**
- 2) **PriortiQueue 사용법**

# 제로 이동 (Move Zeros)

## 설명

정수 배열(nums)이 주어지면 0이 아닌 값은 상대적 순서를 유지하고 nums 모든 0은 끝으로 이동하게 만드세요.

Note) 배열의 복사본을 만들지 않고 작업을 수행해야 합니다.

## 입출력

**Input:** nums = [0,3,2,0,8,5]

**Output:** [3,2,8,5,0,0]

**Input:** nums = [0]

**Output:** [0]

## 문제 Format

```
class Solution {  
    public void solve(int[] nums) { }  
}
```

## 제한사항

$1 \leq \text{nums.length} \leq 10^4$   
 $-2^{31} \leq \text{nums}[i] \leq 2^{31} - 1$

# k번째 제일큰 원소 (Kth Largest Element In An Array)

## 설명

정수 배열 nums와 정수 k가 주어지면 배열에서 k번째로 큰 요소를 반환합니다 .

Note)

k번째로 큰 요소는 정렬후 값에대한 가장 큰 요소가 아닌 유일한(distinctive) 순서 요소입니다.

## 입출력

**Input:** nums = [2,3,1,5,6,4], k = 2

**Output:** 5

**Input:** nums = [3,2,3,1,2,4,5,5,6], k = 4

**Output:** 4

## 문제 Format

```
class Solution {  
    public int solve(int[] nums, int k) { }  
}
```

## 제한사항

$1 \leq k \leq \text{nums.length} \leq 10^4$   
 $-10^4 \leq \text{nums}[i] \leq 10^4$

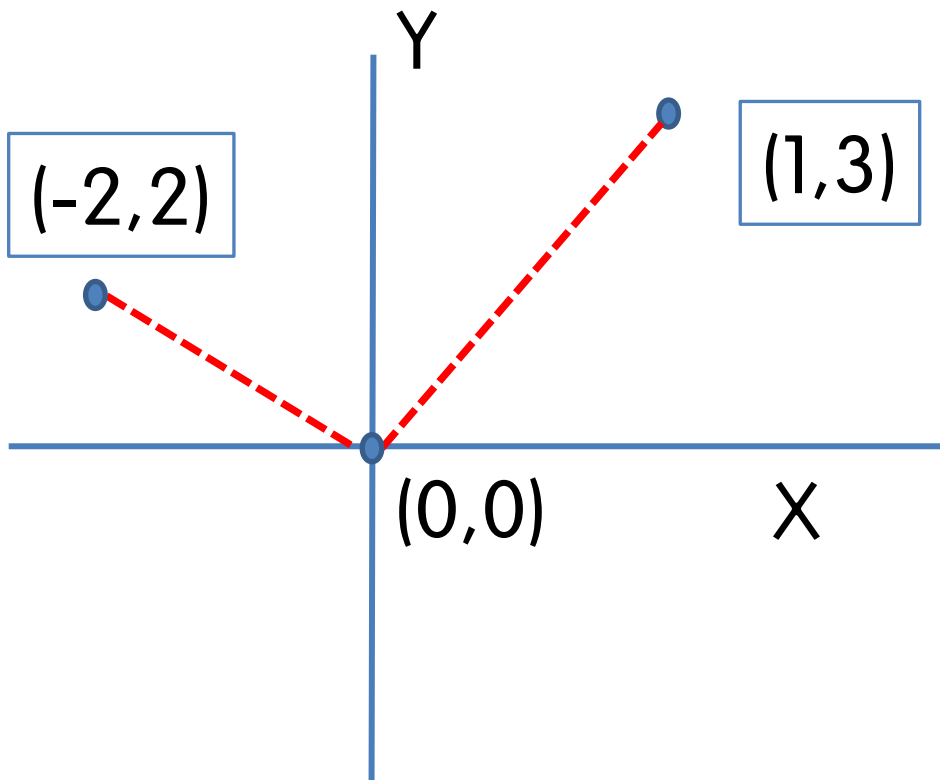
# 원점에 가장 가까운 지점 (K Closest Points to Origin)

## 설명

XY 평면의 한 점 과 정수를 나타내는 배열이 주어지면 원점에 가장 가까운 점을 반환합니다.

XY 평면 에서 두 점 사이의 거리 구하는 공식을 이용하세요.  $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

원점에서 제일 가까운 좌표를 K개의 갯수만큼 리턴하세요



## 입출력

**Input:** points = [[1,3],[-2,2]], k = 1

**Output:** [[-2,2]]

**Explanation:** 원점에서 좌표(1, 3)은 두 점사이 거리구하는 공식을 이용하면 10.

원점에서 좌표(-2, 2)은 두점사이 거리구하는 공식을 이용하면 8.

k = 1 일때는 제일 가까운 좌표 1개만 리턴. [[-2,2]] 리턴

## 원점에 가장 가까운 지점 (K Closest Points to Origin)

**Input:** points = [[3,3],[5,-1],[-2,4]], k = 2

**Output:** [[3,3],[-2,4]]

### 문제 Format

```
class Solution {  
    public int[][] solve(int[][] points, int k) { }  
}
```

### 제한사항

$1 \leq k \leq \text{points.length} \leq 10^4$   
 $-10^4 < x_i, y_i < 10^4$

# 미팅룸 (Meeting Room)

## 설명

미팅 시간 배열이 주어집니다.  $\text{intervals}[i] = [\text{start}_i, \text{end}_i]$   
Intervals 배열을 이용하여 사람들이 모든 회의에 참석할 수 있는지 boolean으로 리턴하세요

## 입출력

**Input:** intervals =  
[ [5,10],[16,20], [0,30]]  
**Output:** false

**Input:** intervals = [[6,10],[1,3]]  
**Output:** true

## 문제 Format

```
class Solution {  
    public boolean solve(int[][] intervals) { }  
}
```

## 제한사항

$0 \leq \text{intervals.length} \leq 10^4$   
 $\text{intervals}[i].\text{length} == 2$   
 $0 \leq \text{start}_i < \text{end}_i \leq 10^6$

# 미팅룸 2(Meeting Room2)

## 설명

미팅 시간 배열이 주어집니다.  $\text{intervals}[i] = [\text{start}_i, \text{end}_i]$   
Intervals배열을 이용하여 사람들이 회의에 참석하려면 몇 개의 회의실이 필요한지 리턴하세요

## 입출력

**Input:** intervals =  
[ [5,10],[16,20], [0,30]]  
**Output:** 2

**Input:** intervals = [[6,10],[1,3]]  
**Output:** 1

## 문제 Format

```
class Solution {  
    public int solve(int[][] intervals) { }  
}
```

## 제한사항

$0 \leq \text{intervals.length} \leq 10^4$   
 $\text{intervals}[i].\text{length} == 2$   
 $0 \leq \text{start}_i < \text{end}_i \leq 10^6$



# interval 병합 (Merge Interval)

## 설명

intervals 배열이 주어지면 겹치는 구간을 병합하여 배열을 반환 합니다 .

## 입출력

**Input:** intervals =

`[[1,4],[2,6],[8,10],[15,18]]`

**Output:** `[[1,6],[8,10],[15,18]]`

**Explanation:** [1,4] [2,6] 오버랩되므로, [1,6]으로 병합시킵니다.

**Input:** intervals = `[[1,5],[5,6]]`

**Output:** `[[1,6]]`

## 문제 Format

```
class Solution {  
    public int[][] solve(int[][] intervals) {}  
}
```

## 제한사항

$1 \leq \text{intervals.length} \leq 10^4$   
 $\text{intervals}[i].\text{length} == 2$   
 $0 \leq \text{start}_i \leq \text{end}_i \leq 10^4$

# 로그 파일의 데이터 재정렬

## 설명

배열로 logs 주어집니다.

각 로그는 공백으로 구분 된 단어 문자열이며 첫 번째 단어는 **식별자** 입니다.

There are two types of logs:

**Letter-logs:** All words (except the identifier) consist of lowercase English letters.

**Digit-logs:** All words (except the identifier) consist of digits.

Reorder these logs so that:

The **letter-logs** come before all **digit-logs**.

The **letter-logs** are sorted lexicographically by their contents.

If their contents are the same, then sort them lexicographically by their identifiers.

The **digit-logs** maintain their relative ordering.

Return the final order of the logs.

## 문제 Format

```
class Solution {  
    public String[] solve (String[] logs) {  
    }  
}
```

### 3. 로그 파일의 데이터 재정렬

#### 설명

**Input:** logs = ["dig1 8 2 3 1",  
"let1 abc cat",  
"dig1 2 5",  
"let2 good dog book",  
"let3 abc zoo" ]

**Output:** ["let1 abc cat",  
"let3 abc zoo",  
"let2 good dog book ",  
"dig1 8 2 3 1",  
"dig1 2 5"]

#### Explanation:

The letter-log contents are all different, so their ordering is "abc cat", "abc zoo", "good dog book".

The digit-logs have a relative order of "dig1 8 2 3 1", "dig1 2 5".

#### 제한사항

$1 \leq \text{logs.length} \leq 100$ ,       $3 \leq \text{logs}[i].\text{length} \leq 100$

All the tokens of logs[i] are separated by a single space.

logs[i] is guaranteed to have an identifier and at least one word after the identifier.