

- 
1. Meeting Room
  2. MoveZeros
  3. Two Sum
  4. Daily Temperature
  5. Merge Interval
  6. Meeting Room2
  7. Jewel and Stone
  8. LicenseKey Formatting
  9. Kclosest
  10. PlusOne

# 전략

---

1. 문제를 정확히 이해한다.
2. 알고리즘을 생각해 본다.

단순 Array, List, Map, LinkedList, Stack, Queue, Set

중간 Comparator, PriorityQueue

고급 DFS, BFS, DP , Greedy

# 문제 1) MeetingRoom

## Problem

Given an array of meeting time intervals consisting of start and end times  $[[s_1, e_1], [s_2, e_2], \dots]$  ( $s_i < e_i$ ), determine if a person could attend all meetings.

Input:  $[[0,30],[5,10],[15,20]]$

Output: false

Input:  $[[7,10],[2,4]]$

Output: true

# MeetingRoom

## Problem

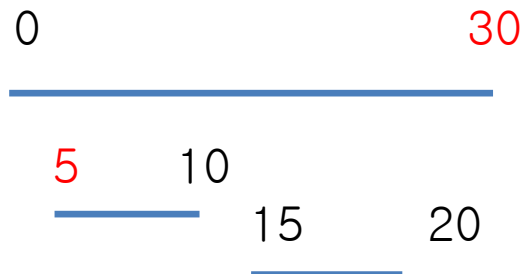
Input: `[[0,30],[5,10],[15,20]]`

Output: `false`

Input: `[[7,10],[2,4]]`

Output: `true`

## Example



1. 소팅
2. `hold.end > cur.start`

# Solution

String

1. 문제를 정확히 이해
2. 알고리즘 정하고 답을 그릇 정한다
3. for 문 돌리기
4. 생각->프로그램(한국말로 생각하고->Java)  
결과를 해석하여 이미지화시킨다

I Can Image

## 문제 2) MoveZeros

### Problem

Given an array `nums`, write a function to move all 0's to the end of it while maintaining the relative order of the non-zero elements.

Example:

Input: [0,1,0,3,12]

Output: [1,3,12,0,0]

Note:

You must do this in-place without making a copy of the array.  
Minimize the total number of operations.

# MoveZeros

## Problem

- 배열 num을 감안할 때 0이 아닌 요소의 상대적인 순서를 유지하면서 모든 0을 끝으로 이동시키는 함수를 작성하십시오

## Example

Input : [0,3,2,0,8,5]

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

1. 값이 0이 아닌 값을 먼저 array에 담는다
2. Index를 기억한다
3. 해당 index 0인 값을 넣는다

## Note



0 3 2 0 8 5



3 2 8 5 0 0

## 2. Solution

Array

1. 문제를 정확히 이해
2. 알고리즘 정하고 답을 그릇 정한다
3. for 문 돌리기

생각->프로그램

결과를 해석을 해서 이미지화시킨다.  
(방번호, 방번호 기억, 0을 제외한다)



## 문제 3) TwoSum

### Problem

Given an array of integers, return indices of the two numbers such that they add up to a specific target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

Example:

Given `nums = [2, 7, 11, 15]`, `target = 9`,

Because `nums[0] + nums[1] = 2 + 7 = 9`,  
return `[0, 1]`.

# Solution

Map, Array

1. 문제를 정확히 이해
2. 알고리즘 정하고 답을 그릇 정한다
3. for 문 돌리기

**Input :**    `int[] nums = {2,8,11,21};`  
             `int target =10;`

**Output :**   `int[] ={1,2}`

생각->프로그램(한국->영어, **한국->Java**)  
결과를 해석을 해서 이미지화시킨다.

Index 0부터 for돌려,  $10-2=8$

맵에 8을 (8, 0)

Index을 돌려서 맵에 있는거랑 매치가 되면  
Index를 구할 수 있어

## 문제 4) Daily Temperature

### Problem

Given a list of daily temperatures  $T$ , return a list such that, for each day in the input, tells you how many days you would have to wait until a warmer temperature. If there is no future day for which this is possible, put 0 instead.

For example, given the list of temperatures  
 $T = [73, 74, 75, 71, 69, 72, 76, 73]$ ,  
your output should be  $[1, 1, 4, 2, 1, 1, 0, 0]$ .

Note: The length of temperatures will be in the range  $[1, 30000]$ .  
Each temperature will be an integer in the range  $[30, 100]$ .

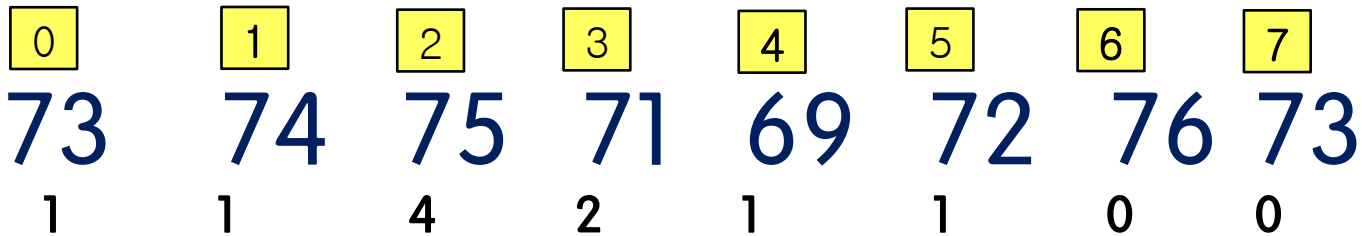
# Daily Temperature

## Problem

```
int[] nums = {73, 74, 75, 71, 69, 72, 76, 73};
```

Output :      1    1    4    2    1    1    0    0

## solution



## 문제 5) MergeInterval

Comparator

### Problem

Given a collection of intervals, merge all overlapping intervals.

### Example

**Input:** `[[1,3],[2,6],[8,10],[15,18]]`

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



## 문제 6) MeetingRoom2

### Problem

Given an array of meeting time intervals consisting of start and end times  $[[s_1, e_1], [s_2, e_2], \dots]$  ( $s_i < e_i$ ), find the minimum number of conference rooms required.

Input:  $[[0,30],[5,10],[15,20]]$

Output: 2

Input:  $[[7,10],[2,4]]$

Output: 1

# MeetingRoom2

## Problem

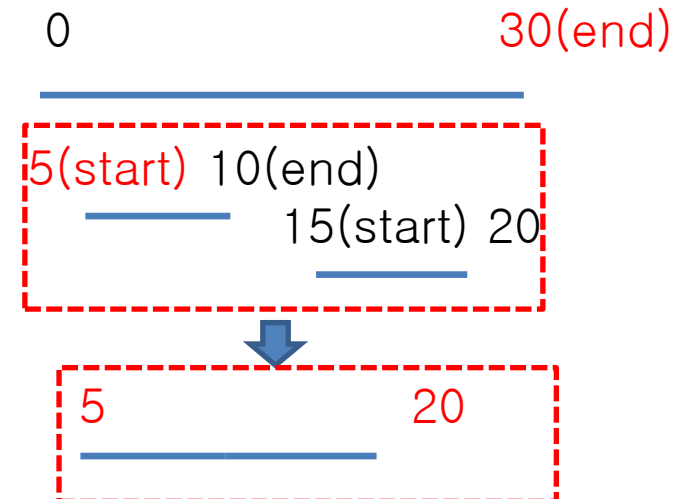
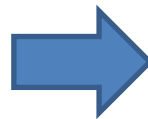
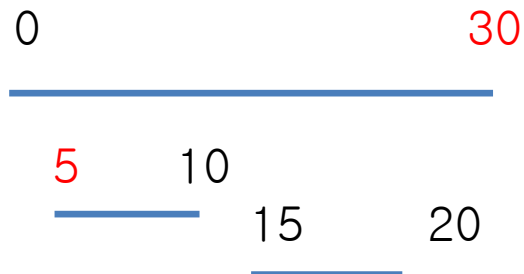
Input: `[[0,30],[5,10],[15,20]]`

Output: 2

Input: `[[7,10],[2,4]]`

Output: 1

## Example



1. 소팅
2. PriorityQueue
3.  $\text{End} \leq \text{start}$  합치기

# Solution

Map, Array

1. 문제를 정확히 이해
2. 알고리즘 정하고 답을 그릇 정한다
3. for 문 돌리기

생각->프로그램(한국말로 생각하고->Java)

결과를 해석하여 이미지화시킨다

1. 클래스(Interval)
2. 리스트화(자료)
3. Comparator(버전)
4. 현재 Start< 전 end , 새롭게 1,6짜리 클래스를 만든다.



## 문제 7) Jewels And Stones

### Problem

You're given strings *J* representing the types of stones that are jewels, and *S* representing the stones you have. Each character in *S* is a type of stone you have. You want to know how many of the stones you have are also jewels.

The letters in *J* are guaranteed distinct, and all characters in *J* and *S* are letters. Letters are case sensitive, so "a" is considered a different type of stone from "A".

**Input:** *J* = "aA", *S* = "aAAbbbb"

**Output:** 3

# Jewels And Stones

## Problem

보석갯수, 스톤갯수, letter 대소문자 구분

## Example

Input: J = "aA", S = "aAAabbbb"

Output: 3

# Solution

Set (중복 X)

1. 문제를 정확히 이해
2. 알고리즘 정하고 답을 그릇 정한다
3. for 문 돌리기
4. 생각->프로그램(한국말로 생각하고->Java)  
결과를 해석하여 이미지화시킨다

I Can Image

1. 보석은 대소문자를 구분해서 갖고 있어야 한다.
2. aA ->2개
3. 스톤에가서 aA가 개별적으로 몇 개이었는지 체크

## 문제 8) LicenseKeyFormatting

### Problem

You are given a license key represented as a string  $S$  which consists only alphanumeric character and dashes. The string is separated into  $N+1$  groups by  $N$  dashes.

Given a number  $K$ , we would want to reformat the strings such that each group contains exactly  $K$  characters, except for the first group which could be shorter than  $K$ , but still must contain at least one character. Furthermore, there must be a dash inserted between two groups and all lowercase letters should be converted to uppercase.

Given a non-empty string  $S$  and a number  $K$ , format the string according to the rules described above.

input

```
String str = "8F3Z-2e-9-w";
```

```
String str = "8-5g-3-J";
```

```
int k =4;
```

Output : 8F3Z-2E9W    8F3Z-2E9W, 8-5G3J

# LicenseKeyFormatting

## Problem

input

```
String str = "8F3Z-2e-9-w";
```

```
String str = "8-5g-3-J";
```

```
int k =4;
```

Output : 8F3Z-2E9W    8F3Z-2E9W, 8-5G3J

## Example

1. 하이픈 제거, 대문자
2. 끝에서 4자리 끊기

String, StringBuffer, StringBuilder 차이점과 장단점.

1. String + 연산이나 concat을 -> 새로운 String객체를 new로 만듦
2. StringBuffer, 멀티쓰레드환경에서는 synchronized
3. StringBuilder asynchronized 싱글쓰레드 환경

# 문제 9) K Closest Points to Origin

PriorityQueue

## Problem

We have a list of points on the plane. Find the K closest points to the origin (0, 0).

(Here, the distance between two points on a plane is the Euclidean distance.)

You may return the answer in any order. The answer is guaranteed to be unique (except for the order that it is in.)

Example 1:

Input: points = [[1,3],[-2,2]], K = 1

Output: [[-2,2]]

Explanation:

The distance between (1, 3) and the origin is  $\sqrt{10}$ .

The distance between (-2, 2) and the origin is  $\sqrt{8}$ .

Since  $\sqrt{8} < \sqrt{10}$ , (-2, 2) is closer to the origin.

We only want the closest K = 1 points from the origin, so the answer is just [[-2,2]].

# K Closest Points to Origin

PriorityQueue

## Problem

Input: points =  $[[1,3],[-2,2]]$ ,  $K = 1$

Output:  $[-2,2]$

$(0,0)(1, 3)$  and the origin is  $\sqrt{10}$   $1+9$

$(0,0)(-2, 2)$  and the origin is  $\sqrt{8}$   $4+4$

## 문제 10) Plus One

### Problem

Given a non-empty array of digits representing a non-negative integer, plus one to the integer.

The digits are stored such that the most significant digit is at the head of the list, and each element in the array contain a single digit.

You may assume the integer does not contain any leading zero, except the number 0 itself.

Example 1:

Input: [1,2,3]

Output: [1,2,4]

Explanation: The array represents the integer 123.



# Plus One

## Problem

Input: [1,2,3]

Output: [1,2,4]

Input: [9,9,9]

Output: [1,0,0,0]

## Example

1. 뒷자리부터 9가 있는지 체크
2. input[0], input[1], input[2]
3. input[2] = 0;
4. input[0] == 1 , 새로운 int[] 의 크기를 1증가
5. 새로운 배열 res[0]=1

1. 9가 아닌경우는 해당 자리수의 값을 +1하면 끝