1. K 번째수

문제 설명

배열 array 의 i 번째 숫자부터 j 번째 숫자까지 자르고 정렬했을 때, k 번째에 있는 수를 구하려 합니다.

예를 들어 array 가 [1, 5, 2, 6, 3, 7, 4], i = 2, j = 5, k = 3 이라면

1. array 의 2 번째부터 5 번째까지 자르면 [5, 2, 6, 3]입니다.
2. 1 에서 나온 배열을 정렬하면 [2, 3, 5, 6]입니다.
3. 2 에서 나온 배열의 3 번째 숫자는 5 입니다.

배열 array, [i, j, k]를 원소로 가진 2 차원 배열 commands 가 매개변수로 주어질 때, commands 의 모든 원소에 대해 앞서 설명한 연산을 적용했을 때 나온 결과를 배열에 담아 return 하도록 solution 함수를 작성해주세요.

제한사항

- array 의 길이는 1 이상 100 이하입니다.
- array 의 각 원소는 1 이상 100 이하입니다.
- commands 의 길이는 1 이상 50 이하입니다.
- commands 의 각 원소는 길이가 3 입니다.

입출력 예

array	commands	return
[1, 5, 2, 6, 3, 7, 4]	[[2, 5, 3], [4, 4, 1], [1, 7, 3]]	[5, 6, 3]

입출력 예 설명

[1, 5, 2, 6, 3, 7, 4]를 2 번째부터 5 번째까지 자른 후 정렬합니다. [2, 3, 5, 6]의 세 번째 숫자는 5 입니다.
[1, 5, 2, 6, 3, 7, 4]를 4 번째부터 4 번째까지 자른 후 정렬합니다. [6]의 첫 번째 숫자는 6 입니다.
[1, 5, 2, 6, 3, 7, 4]를 1 번째부터 7 번째까지 자릅니다. [1, 2, 3, 4, 5, 6, 7]의 세 번째 숫자는 3 입니다.

```
import java.util.*;
class Solution {
    public int[] solution(int[] array, int[][] commands) {
        int[] answer = new int[commands.length];
```

```
        for(int i =0; i<commands.length; i++) {
            int[] temp = Arrays.copyOfRange(array, commands[i][0
]-1,commands[i][1]);
            Arrays.sort(temp);
            answer[i]=temp[commands[i][2]-1];
        }
        return answer;
    }
}
```

```
class Solution {
    public int[] solution(int[] array, int[][] commands) {
        int n = 0;
        int[] ret = new int[commands.length];

        while(n < commands.length){
            int m = commands[n][1] - commands[n][0] + 1;

            if(m == 1){
                ret[n] = array[commands[n++][0]-1];
                continue;
            }

            int[] a = new int[m];
            int j = 0;
            for(int i = commands[n][0]-
1; i < commands[n][1]; i++)
                a[j++] = array[i];

            sort(a, 0, m-1);

            ret[n] = a[commands[n++][2]-1];
        }

        return ret;
    }

    void sort(int[] a, int left, int right){
        int pl = left;
        int pr = right;
        int x = a[(pl+pr)/2];

        do{
            while(a[pl] < x) pl++;
            while(a[pr] > x) pr--;
            if(pl <= pr){
                int temp = a[pl];
                a[pl] = a[pr];
                a[pr] = temp;
```

```
                pl++;
                pr--;
            }
        }while(pl <= pr);

        if(left < pr) sort(a, left, pr);
        if(right > pl) sort(a, pl, right);
    }
}
```

https://sas-study.tistory.com/3

2. 가장 큰 수

문제 설명

0 또는 양의 정수가 주어졌을 때, 정수를 이어 붙여 만들 수 있는 가장 큰 수를 알아내 주세요.

예를 들어, 주어진 정수가 [6, 10, 2]라면 [6102, 6210, 1062, 1026, 2610, 2106]를 만들 수 있고, 이중 가장 큰 수는 6210 입니다.

0 또는 양의 정수가 담긴 배열 numbers 가 매개변수로 주어질 때, 순서를 재배치하여 만들 수 있는 가장 큰 수를 문자열로 바꾸어 return 하도록 solution 함수를 작성해주세요.

제한 사항

- numbers 의 길이는 1 이상 100,000 이하입니다.
- numbers 의 원소는 0 이상 1,000 이하입니다.
- 정답이 너무 클 수 있으니 문자열로 바꾸어 return 합니다.

입출력 예

Numbers	return
[6, 10, 2]	6210
[3, 30, 34, 5, 9]	9534330

```
public String solution(int[] numbers) {
	String answer = new String();
	/** 1 **/
	String str_numbers[] = new String[numbers.length];

	/** 2 **/
	for(int i=0; i<str_numbers.length; i++) {
		str_numbers[i] = String.valueOf(numbers[i]);
	}

	/** 3 **/
	Arrays.sort(str_numbers, new Comparator<String>() {
		@Override
		public int compare(String o1, String o2) {
			return (o2+o1).compareTo(o1+o2);
		}
	});

	/** 4 **/
```

```
        if(str_numbers[0].startsWith("0")) {
                answer += "0";
        } else {
                for(int j=0; j<str_numbers.length; j++) {
                        answer += str_numbers[j];
                }
        }

        return answer;
}
```

https://m.blog.naver.com/PostView.nhn?blogId=yongyos&logNo=221580402785&categoryNo=39&proxyReferer=https:%2F%2Fwww.google.com%2F

3. H-Index

문제 설명

H-Index 는 과학자의 생산성과 영향력을 나타내는 지표입니다. 어느 과학자의 H-Index 를 나타내는 값인 h 를 구하려고 합니다. 위키백과[1]에 따르면, H-Index 는 다음과 같이 구합니다.

어떤 과학자가 발표한 논문 `n` 편 중, `h` 번 이상 인용된 논문이 `h` 편 이상이고 나머지 논문이 h 번 이하 인용되었다면 `h` 의 최댓값이 이 과학자의 H-Index 입니다.

어떤 과학자가 발표한 논문의 인용 횟수를 담은 배열 citations 가 매개변수로 주어질 때, 이 과학자의 H-Index 를 return 하도록 solution 함수를 작성해주세요.

제한사항

- 과학자가 발표한 논문의 수는 1 편 이상 1,000 편 이하입니다.
- 논문별 인용 횟수는 0 회 이상 10,000 회 이하입니다.

입출력 예

citations	return
[3, 0, 6, 1, 5]	3

입출력 예 설명

이 과학자가 발표한 논문의 수는 5 편이고, 그중 3 편의 논문은 3 회 이상 인용되었습니다. 그리고 나머지 2 편의 논문은 3 회 이하 인용되었기 때문에 이 과학자의 H-Index 는 3 입니다.

※ 공지 - 2019 년 2 월 28 일 테스트 케이스가 추가되었습니다.

```
import java.util.*;

public class Solution {
    public int solution(int[] citations) {
        int answer = 0;
        Arrays.sort(citations);

        for (int i = 0; i < citations.length; i++) {
            int h = citations.length - i;

            if (citations[i] >= h) {
                answer = h;
                break;
```

```
                }
            }

            return answer;
        }
}
```

https://ju-nam2.tistory.com/74