# 회의실배정 실패

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **시간 제한** | **메모리 제한** | **제출** | **정답** | **맞은 사람** | **정답 비율** |
| 2 초 | 128 MB | 14102 | 3756 | 2853 | 29.142% |

## 문제

한 개의 회의실이 있는데 이를 사용하고자 하는 N개의 회의들에 대하여 회의실 사용표를 만들려고 한다. 각 회의 I에 대해 시작시간과 끝나는 시간이 주어져 있고, 각 회의가 겹치지 않게 하면서 회의실을 사용할 수 있는 최대수의 회의를 찾아라. 단, 회의는 한번 시작하면 중간에 중단될 수 없으며 한 회의가 끝나는 것과 동시에 다음 회의가 시작될 수 있다. 회의의 시작시간과 끝나는 시간이 같을 수도 있다. 이 경우에는 시작하자마자 끝나는 것으로 생각하면 된다.

## 입력

첫째 줄에 회의의 수 N(1 ≤ N ≤ 100,000)이 주어진다. 둘째 줄부터 N+1 줄까지 각 회의의 정보가 주어지는데 이것은 공백을 사이에 두고 회의의 시작시간과 끝나는 시간이 주어진다. 시작 시간과 끝나는 시간은 231-1보다 작거나 같은 자연수 또는 0이다.

## 출력

첫째 줄에 최대 사용할 수 있는 회의 수를 출력하여라.

## 예제 입력 1 복사

11  
1 4  
3 5  
0 6  
5 7  
3 8  
5 9  
6 10  
8 11  
8 12  
2 13  
12 14

## 예제 출력 1 복사

4

## 힌트

(1,4), (5,7), (8,11), (12,14) 를 이용할 수 있다.

## 출처

* 빠진 조건을 찾은 사람: [andy627](https://www.acmicpc.net/user/andy627) [breakun](https://www.acmicpc.net/user/breakun)
* 데이터를 추가한 사람: [kimchangyoung](https://www.acmicpc.net/user/kimchangyoung) [rosedskim](https://www.acmicpc.net/user/rosedskim)

## 알고리즘 분류

[보기](https://www.acmicpc.net/problem/1931#)

## 메모

123

answer - 1 틀림

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collections;

import java.util.Comparator;

import java.util.StringTokenizer;

public class T\_1931 {

public static void main(String[] args) throws IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

StringTokenizer st = new StringTokenizer(br.readLine());

int line = Integer.parseInt(st.nextToken());

ArrayList<Integer[]> list = new ArrayList<Integer[]>();

for(int i=0;i<line;i++){

st = new StringTokenizer(br.readLine());

Integer[] data = new Integer[2];

data[0] = Integer.parseInt(st.nextToken());

data[1] = Integer.parseInt(st.nextToken());

list.add(data);

}

Collections.sort(list, new Comparator<Integer[]>() {

@Override

public int compare(Integer[] s1, Integer[] s2) {

return s1[0].compareTo(s2[0]);

}

});

int count=1;

Integer[] compare = list.get(list.size()-1);

for(int i=list.size()-2;i>=0;i--){

if(compare[0] < list.get(i)[0]){

compare[0] = list.get(i)[0];

}else if(compare[0] > list.get(i)[1]){

count++;

compare = list.get(i);

}

}

System.out.println(count);

}

}