

BOJ: 1931번 회의실 배정 풀이

문제 설명

문제

한 개의 회의실이 있는데 이를 사용하고자 하는 N 개의 회의에 대하여 회의실 사용표를 만들려고 한다. 각 회의 i 에 대해 시작시간과 끝나는 시간이 주어지고, 각 회의가 겹치지 않게 하면서 회의실을 사용할 수 있는 회의의 최대 개수를 찾아보자. 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 ($1 \leq N \leq 100,000$)이 주어진다. 둘째 줄부터 $N+1$ 줄까지 각 회의의 정보가 주어지는데 이것은 공백을 사이에 두고 회의의 시작시간과 끝나는 시간이 주어진다. 시작 시간과 끝나는 시간은 $2^{31} - 1$ 보다 작거나 같은 자연수 또는 0이다.

출력

첫째 줄에 최대 사용할 수 있는 회의의 최대 개수를 출력한다.

접근 방법

처음엔 그리디한 접근을 해야겠다는 생각으로 시작 시간이 빠른 회의를 선택했지만, 이 접근 방법은 올바른 답을 유도하지 못했다. 힌트를 보고 끝나는 시간이 빠른 회의를 선택하는 것이 올바른 선택인 것을 알았다. 끝나는 시간이 같은 경우, 시작 시간이 가장 빠른 것을 먼저 선택하는 것이 회의의 개수를 최대로 하는 방법이다.

주요 코드

- 입력 받은 배열을 끝나는 시간을 기준으로 오름차순 정렬

```
int N;
cin >> N;
vector<pair<int, int>> inputs(N);
for (int i = 0; i < N; i++) {
    cin >> inputs[i].first >> inputs[i].second;
}
```

```
sort(inputs.begin(), inputs.end(), [](auto a, auto b) {
    if (a.second == b.second)
        return a.first < b.first;
    return a.second < b.second;
});
```

- 조건에 맞는 회의를 찾으면 cnt 증가

```
int cnt = 1;
int curTime = inputs[0].second;

for (int i = 1; i < N; i++) {
    if (curTime <= inputs[i].first) { // (현재 시간 <= 다음 회의 시작 시간)이면 회의
        curTime = inputs[i].second;
        cnt++;
    }
}
```