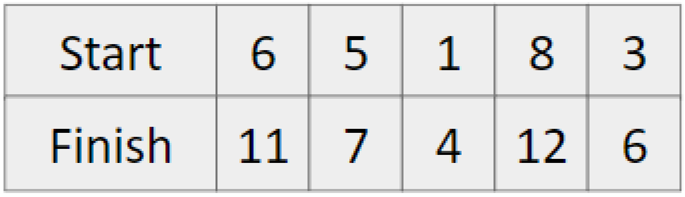
Activity Selection Problem

Problem

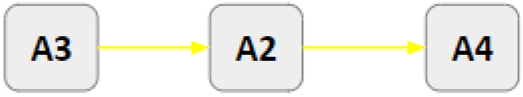
We are given n activities with their start and finish times. We have to select the maximum number of activities such that no two selected activities overlap.

Example



Start	6	5	1	8	3
Finish	11	7	4	12	6

We can select activities in this order

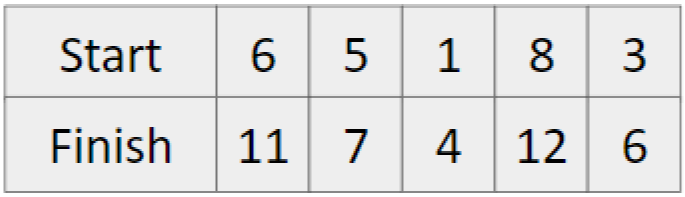


Approach

1. Sort all the activities by their finish times
2. Greedily select the first activity and jump on the next.
3. If the starting time of current activity is greater than the ending time of previously selected activity, then take it otherwise ignore it.

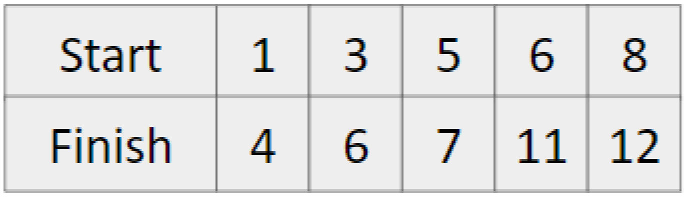
Dry Run

Original start and end times



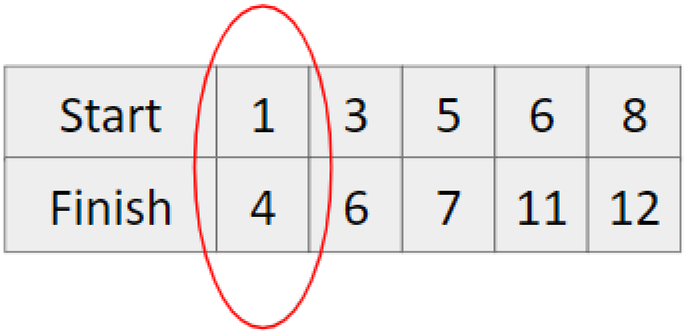
Start	6	5	1	8	3
Finish	11	7	4	12	6

After sorting in increasing order

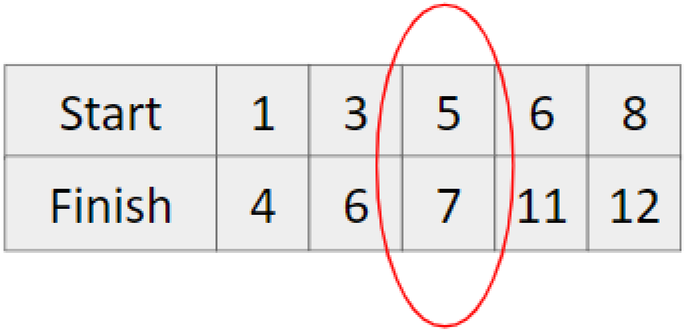


Start	1	3	5	6	8
Finish	4	6	7	11	12

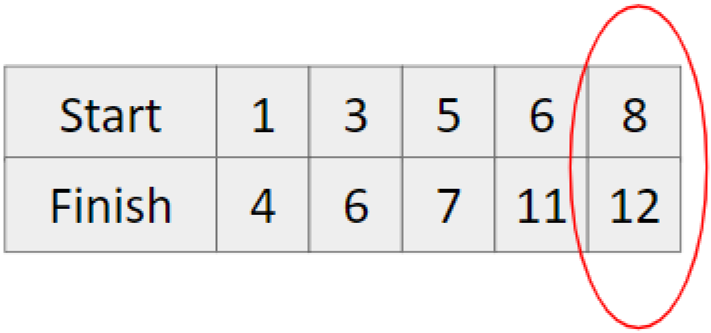
We can select activities in following order



Start	[✓ 1]	3	5	6	8
Finish	[✓ 4]	6	7	11	12



Start	1	3	[✓ 5]	6	8
Finish	4	6	[✓ 7]	11	12



Start	1	3	5	6	[✓ 8]
Finish	4	6	7	11	[✓ 12]

Code

```
#include<bits/stdc++.h>
using namespace std;
#define int long long

bool compare(pair<int,int> t1, pair<int,int> t2) {
    if(t1.second == t2.second) {
        t1.first < t2.first;
    }

    return t1.second < t2.second;
}

signed main() {
    int n; cin >> n;

    vector<pair<int,int>> times;
    for(int i=0; i<n; i++) {
        int st, fn;
        cin >> st >> fn;

        times.push_back({st,fn});
    }

    sort(times.begin(), times.end(), compare);

    int ans = 1;
    int previousEndTime = times[0].second;

    for(int i=1; i<n; i++) {
        if(times[i].first >= previousEndTime) {
            ans++;
            previousEndTime = times[i].second;
        }
    }

    cout << ans << endl;
    return 0;
}
```