

255.Meeting Rooms II

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.

For example,

Given $[[0, 30],[5, 10],[15, 20]]$,

return 2.

```
//C++
//Author:DemonMikalis(ZZW)
#include <iostream>
#include <sstream>
#include <vector>
#include <algorithm>
using namespace std;
struct Interval
{
    int start;
    int end;
};

int meetingRooms2(vector<Interval> &intervals)
{
    vector<int> s(intervals.size(),0);
    vector<int> e(intervals.size(),0);
    for(int k=0;k<(int)intervals.size();k++)
    {
        Interval tmp = intervals[k];
        s[k] = tmp.start;
        e[k] = tmp.end;
    }
    sort(s.begin(),s.end());
    sort(e.begin(),e.end());
    int room=0;int available=0;int eindex=0;
    for(int i=0;i<(int)intervals.size();i++)
    {
        while(e[eindex]<=s[i])
        {
            eindex++;
            available++;
        }
        if (available>0)
            available--;
        else{
            room++;
        }
    }
    return room;
}

int main(int argc,char *argv[])
{
    /* [[0, 30],[5, 10],[15, 20]],
    */
    int array[3][2] = {{0,30},{5,10},{15,20}};
    vector<Interval> intervals;
    for(int i=0;i<3;i++)
    {
        Interval z;
        z.start=array[i][0];
```

```
        z.end=array[i][1];
        intervals.push_back(z);
    }
    //cout << intervals[0].start << intervals[1].start << endl;
    int room = meetingRooms2(intervals);
    cout << room << endl;
    return 0;
}
```