

Week 2 Programming Assignment - Moving Intervals (C++)

Credits : <https://www.codechef.com/users/bhargavitejura>

Explanation : <https://www.codechef.com/viewsolution/78726986>

Document uploaded to: <https://github.com/sujallimje/NPTEL-DAA-ASSIGNMENT-2024>

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
bool canAdjust(vector<pair<int,int>> v, int cakes, int fightIdx){
```

```
    int l = v[fightIdx].first, h = v[fightIdx].second; // stored the fighting kids indexes cuz im gonna give them a new value
```

```
    v[fightIdx].first = -1; // stored -1 in them cuz when im inside the loop and trying to find gap the fighting kids wont interfere
```

```
    v[fightIdx].second = -1;
```

```
    v.push_back({0,0});
```

```
    v.push_back({cakes+1, cakes+1});
```

```
    int gapIdx = -1;
```

```
    sort(v.begin(), v.end());
```

```
    for(int i = 2; i < v.size(); i++){ // i is 2 cuz first 2 are dummy(-1,0)
```

```
    /* 5 2 1
```

```
    -1 -1 (2, 2)
```

```
    0 0
```

```
    2 5
```

```
    6 6
```

```
    FINDING GAP
```

```
    lets say i is 2 (for now), we're doing
```

```
    (v[i].first - v[i-1].second) which is also
```

```
    2-0 which is 2 but if u observe u will see that theres only 1 gap
```

and not 2 gaps so did $(v[i].first - v[i-1].second) - 1$

same in $(h-l)+1$

in the if condition im trying to find is the curr gap enough to fit the fighting kid's cake range or not

```
*/
    if((v[i].first - v[i-1].second)-1 >= (h-l)+1){
        gapIdx = i;
    }
}
if(gapIdx == -1){
    return false;
}else{
    v[0].first = v[gapIdx-1].second+1; // (0+1=1) added 1 to allow eating of the next cake after prev
    child finishes
    v[0].second = v[0].first+(h-l); // (1+(1-1)=1) To eat same number of cakes (h-l)
    // 5 2 1
    // 0 0
    // 1 1(2,2)
    // 2 5
    // 6 6
    sort(v.begin(), v.end());
    for(int i = 1; i < v.size(); i++){ // i=1 cuz in the loop im doing v[i-1].second so if i is at 0 index then
    WA will come
        if(v[i-1].second >= v[i].first){ // checking if there's any overlapping, v is sorted so no issue will
        come
            return false;
        }
    }
    return true;
}
int main(){
```

```

ios_base::sync_with_stdio(0);
cin.tie(0);
int t=1;

while(t--){
    int cakes,n,k;
    cin>>cakes>>n>>k;
    vector<pair<int, int>> v(n);
    for(int i = 0; i < n; i++){
        cin>>v[i].first>>v[i].second;
    }
    sort(v.begin(), v.end());
    int fightIdx = -1;
    bool fight = false;
    for(int i = 1; i < n; i++){ // i=1 cuz in the loop im doing v[i-1].second so if i is at 0 index then WA
will come
        if(v[i-1].second >= v[i].first){ // checking if there's any overlapping, v is sorted so no issue will
come
            fightIdx = i;
            fight = true;
            break;
        }
    }
    if(!fight){
        cout<<"Good\n";
    }else if(fight && k == 0){ // we can only persuade 1 child but if 2 or more children are fighting we
cant make them stop fighting
        cout<<"Bad\n";
    }else if(fight && k == 1){
        if(canAdjust(v, cakes, fightIdx) || canAdjust(v, cakes, fightIdx-1)){
            cout<<"Good\n";
        }else{

```

```
        cout<<"Bad\n";  
    }  
}  
}  
return 0;  
}
```