SJTU OJ 1247 解题报告

F1503024　515030910585　金陆骅

2017 年 5 月 23 日

目录

1. Description

玩过魔兽争霸 3 的同学们都知道暗夜精灵有一个英雄叫老鹿，他有一个技能是召唤树人。在这个过程中，他会把一定范围内的树变成树人。现在我们假设老鹿面前有一排树，他们的编号是 0,1,2,...,L。假设技能覆盖的范围由两个整数给出，在这个范围内的树会全部变成树人。我们现在要知道在老鹿释放了 M 次召唤树人的技能后还剩下几棵树。

2. Input Format

第一行：输入两个整数 L，M。L 代表树木最大的编号，M 代表老鹿释放召唤树人技能的次数。L 和 M 之间用一个空格隔开。

接下来的 M 行：每行都有两个整数，这两个整数由一个空格隔开，表示老鹿技能的释放范围（两端的树也在范围内）。

3. Output Format

输出一行，这一行只有一个整数，表示在释放技能之后还剩余的树的数目。

4. Solution Method

根据远坂家族“秉持优雅”的家训，太过暴力的方法显然是无法接受的，因此我们不能使用每次都去检查技能范围内的树是否已经不存在的方法，那如何来简化算法呢？

显然，当各技能释放区域没有重叠部分时，可以直接减去各区域树的数量以得到剩下树的数量。那么我们要做的，就是将有重叠的区域合并成一个完整的区域。方法也很简单，假定区域一开始位置在区域二之前，只要比较区域一与区域二结束的位置，将较大的位置作为新的区域的结束位置即可。

为了保证区域开始位置的有序性，只需根据其开始位置进行一次排序即可。

5. Source Code

```
#include<iostream>
using namespace std;

struct range {
    int begin; //开始位置
    int end;   //结束位置
};

void quickSort(range s[], int l, int r) //以开始位置从小到大的快速排序
{
    if (l< r)
    {
        int i = l, j = r;
        range x = s[l];
        while (i < j)
        {
            while (i < j && s[j].begin >= x.begin)
                j--;
            if (i < j)
                s[i++] = s[j];
            while (i < j && s[i].begin < x.begin)
                i++;
            if (i < j)
                s[j--] = s[i];
        }
        s[i] = x;
        quickSort(s, l, i - 1);
```

```
        quickSort(s, i + 1, r);
    }
}

int main()
{
    int i,L,m,res,b,e;
    cin >> L >> m;
    res = L + 1; //剩余树数
    range *skill = new range[m];

    for (i = 0;i < m;++i)
    {
        cin >> skill[i].begin >> skill[i].end;
    }

    quickSort(skill,0,m - 1);
    b = skill[0].begin;
    e = skill[0].end;

    for (i = 1;i < m;++i)
    {
        if(skill[i].begin <= e)//有重叠部分，更新结束位置
            e = e > skill[i].end? e:skill[i].end;
        else//无重叠部分，更新树数、开始位置、结束位置
        {
            res -= e - b + 1;
            b = skill[i].begin;
            e = skill[i].end;
        }
    }
    res -= e - b + 1;

    cout << res << endl;

    return 0;
}
```