

种树 (trees.cpp) //P1250

【问题描述】

一条街的一边有几座房子，因为环保原因，居民们想要在路边种些树。路边的地区被分割成块，并被编号成 $1..N$ 。每个部分为一个单位尺寸大小，并且最多可以种植一棵树。每个居民都想在门前种些树，并要指定三个号码 B 、 E 、 T ，这三个数表示该居民想在 B 和 E 之间最少种植 T 棵树。当然， $B \leq E$ 。居民们必须记住在指定区域不能种植多于区域地块数的树，所以 $T \leq E - B + 1$ 。居民们想种植树的区域可以交叉，你的任务是求出能满足所有要求的最少的树的数量和位置。

【输入】(trees.in)

第一行包含数据 N ，表示区域的个数 ($0 < N \leq 30000$)；

第二行包含 H ，房子的数目 ($0 < H \leq 5000$)；

下面的 H 行描述了居民们的需要： $B \ E \ T$ ($0 < B \leq E \leq 30000$, $T \leq E - B + 1$)。

【输出】(trees.out)

输出文件的第一行为树的数目，第二行为所有树的位置，相邻两数之间用一个空格隔开。

【样例】

trees.in:

9

4

1 4 2

4 6 2

8 9 2

3 5 2

trees.out:

5

1 4 5 8 9

【参考程序】

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

struct node{
    int b, e, t;
}h[5005];

int a[30005];    //打勾用的数组

bool cmp(node a, node b){
    if (a.e != b.e)
        return a.e < b.e;
    else
        return a.t > b.t;
}
```

```

int main(){
    int n, q;
    cin >> n >> q;

    for (int i=0; i<q; i++){
        cin >> h[i].b >> h[i].e >> h[i].t;
    }

    sort(h, h+q, cmp);

    for (int i=0; i<q; i++)
    {
        int cnt=0;
        for (int j=h[i].b; j<=h[i].e; j++) cnt += a[j];

        if (cnt < h[i].t)
        {
            int x = h[i].t - cnt;    // 4-2=2
            int j = h[i].e;
            while (x){
                if (a[j]!=1) {
                    a[j]=1;
                    x--;
                }
                j--;
            }
        }
    }

    return 0;
}

```