

# 题意

区间覆盖，每次用一种给定颜色覆盖一条线段  
最后查询每一种颜色有几条线段

# 思路

线段  $[x,y]$ ， $x++$ 后，就可以看成覆盖点 $x$ 到 $y$

update

方式和贴海报相同

query

先左后右的方式递归会让我们，从左到右遍历到所有线段

如果当前区间有标记，就把这个颜色加入vector尾部

如果到叶子结点，仍然没标记，把-1加入vector尾部

遍历vector，连续相同的颜色点视作一个点，统计每种颜色出现过几次

# 代码

```
1  #include<bits/stdc++.h>
2  #define For(i,a,b) for(int i=(a); i<=(b) ; i++)
3  #define _For(i,a,b) for(int i=(a); i>=(b) ; i--)
4  #define Memset(a,b); memset((a),(b),sizeof((a)));
5  #define Cin(a) scanf("%d",&(a))
6  #define Cinc(a) scanf(" %c",&(a))
7  #define Cins(a) scanf("%s",&(a))
8  #define Cout(a,b); printf("%d",(a));printf(b);
9  #define Coutc(a,b); printf("%c",(a));printf(b);
10 #define Coutss(a,b); printf("%s",(a));printf(b);
11 using namespace std;
12 typedef long long LL;
13 typedef unsigned long long ULL;
14 typedef long double LDB;
15 inline int readint() {int x;cin>>x;return x;}
16 int n;
17 int tree[40000];
18 int x[10000],y[10000],c[10000];
19 inline void pushdown(int o)
20 {
21     tree[o<<1] = tree[o<<1|1] = tree[o];
22     tree[o] = -1;
```

```

23 }
24 void update(int o,int l,int r,int L,int R,int d)
25 {
26     if(l>=L && r<=R)
27     {
28         tree[o] = d;
29         return;
30     }
31     if(l==r) return;
32     if(tree[o]!=-1) pushdown(o);
33     int M = (l+r)>>1;
34     if(M>=L) update(o<<1,l,M,L,R,d);
35     if(M+1<=R) update(o<<1|1,M+1,r,L,R,d);
36 }
37 vector<int>ans;
38 void query(int o,int l,int r)
39 {
40     if(tree[o]!=-1)
41     {
42         ans.push_back(tree[o]);
43         return;
44     }
45     if(l == r)
46     {
47         ans.push_back(-1);
48         return;
49     }
50     int M = (l+r)>>1;
51     query(o<<1,l,M);
52     query(o<<1|1,M+1,r);
53 }
54 int main()
55 {
56     while(scanf("%d",&n)!=EOF)
57     {
58         memset(tree,-1,sizeof(tree));
59         int m = -1,cmax=-1;
60         For(i,1,n)
61         {
62             Cin(x[i]);Cin(y[i]);Cin(c[i]);
63             x[i]++;
64             if(y[i] > m) m = y[i];
65             if(c[i]>cmax) cmax = c[i];
66         }
67         For(i,1,n)
68             update(1,1,m,x[i],y[i],c[i]);

```

```
69     ans.clear();
70     query(1,1,m);
71     memset(c,0,sizeof(c));
72     if(ans[0]!=-1)
73         c[ans[0]] ++;
74     for(int i=1;i<ans.size();i++)
75     {
76         if(ans[i]!=ans[i-1]&&ans[i]!=-1)
77             c[ans[i]]++;
78     }
79     for(int i=0;i<=cmax;i++)
80     {
81         if(c[i] == 0)continue;
82         printf("%d %d\n",i,c[i]);
83     }
84     putchar('\n');
85 }
86 }
```