

题意

下标0到n-1，初始全为0

两种操作：

1 a f 从位置a开始往后，每遇到0就将它变为1，直到改变了f个或者到末尾，输出改变的第一个值和最后一个值的下标

2 x y 把[x,y]所有的值置为0，并输出总共改变了多少个值

分析

线段树维护当前区间还有多少空位tree[o]

lazy标记当前区间全被置空（0）或置满（1），或无标记（-1）

操作1：

1. 先判断从a位置到末尾总共有多少个空位k

区间查询和

2. 输出左值l = 第一个空位下标，输出右值r = 第min(f,k) 个空位下标、

3. 将[l,r] 区间全部置为1即tree[o] = 0

操作2：

直接将所给区间全部置为0即tree[o] = 区间长度（常规操作！）

返回值为改变的tree[o]

pushdown:

如果标记存在

如果标记为1：将tree左右区间全部赋值为0，表示全满

如果标记为0：将tree左右区间全部赋值为区间长度，表示全空

清空当前区间的标记：lazy[o] = -1;

代码

```
1 #include<iostream>
2 #include<cstring>
3 #include<cstdio>
4 #include<algorithm>
5 #define For(i,a,b) for(int i=(a); i<=(b) ; i++)
6 #define _For(i,a,b) for(int i=(a); i>=(b) ; i--)
7 #define Memset(a,b); memset((a),(b),sizeof((a)));
8 #define Cin(a); scanf("%d",&(a));
9 #define Cinc(a); scanf(" %c",&(a));
10 #define Cins(a); scanf("%s",&(a));
```

```

11 #define Cout(a,b); printf("%d",(a));printf(b);
12 #define Coutc(a,b); printf("%c",(a));printf(b);
13 #define Couts(a,b); printf("%s",(a));printf(b);
14 using namespace std;
15 typedef long long LL;
16 typedef unsigned long long ULL;
17 typedef long double LDB;
18 inline int readint() {int x;cin>>x;return x;}
19 int n,m;
20 int cmd,x,y;
21 int tree[200005],lazy[200005];
22 void pushdown(int o,int l,int r)
23 {
24     lazy[o<<1] = lazy[o<<1|1] = lazy[o];
25     if(lazy[o] == 1)
26         tree[o<<1] = tree[o<<1|1] = 0;
27     else{
28         int M = (l+r)>>1;
29         tree[o<<1] = M-l+1;
30         tree[o<<1|1] = r-M;
31     }
32     lazy[o] = -1;
33 }
34 void build(int o,int l,int r)
35 {
36     if(l==r) {
37         tree[o] = 1;
38         return;
39     }
40     int M = (l+r)>>1;
41     build(o<<1,l,M);
42     build(o<<1|1,M+1,r);
43     tree[o] = tree[o<<1]+tree[o<<1|1];
44 }
45 void update(int o,int l,int r,int L,int R)
46 {
47     if(l >=L && r<=R){
48         tree[o] = 0;
49         lazy[o] = 1;//该段为满
50         return ;
51     }
52     if(lazy[o] != -1) pushdown(o,l,r);
53     int M = (l+r)>>1;
54     if(M>=L) update(o<<1,l,M,L,R);
55     if(M+1<=R) update(o<<1|1,M+1,r,L,R);
56     tree[o] = tree[o<<1]+tree[o<<1|1];

```

```

57 }
58 int update2(int o,int l,int r,int L,int R)
59 {
60     if(l>=L && r<=R){
61         int temp= tree[o];
62         tree[o] = r-l+1;
63         lazy[o] = 0; //该段为空
64         return tree[o] - temp;
65     }
66     if(lazy[o] != -1 ) pushdown(o,l,r);
67     int M = (l+r)>>1;
68     int ans = 0;
69     if(M>=L) ans+=update2(o<<1,l,M,L,R);
70     if(M+1<=R) ans+=update2(o<<1|1,M+1,r,L,R);
71     tree[o] = tree[o<<1] + tree[o<<1|1];
72     return ans;
73 }
74 int query(int o,int l,int r,int L,int R)
75 {
76     if(l>R || r< L) return 0;
77     if(L<=l && r<=R) return tree[o];
78     if(lazy[o] !=-1) pushdown(o,l,r);
79     int M = (l+r)>>1;
80     int ans = 0;
81     if(M>=L) ans+=query(o<<1,l,M,L,R);
82     if(M+1<=R) ans+=query(o<<1|1,M+1,r,L,R);
83     return ans;
84 }
85 int findd(int o,int l,int r,int L,int R,int num)//找区间L到R间的第num个空位
86 {
87     if(l == r) return l;
88     if(lazy[o]!=-1) pushdown(o,l,r);
89     int M = (l+r)>>1;
90     int k = query(o<<1,l,M,L,R); // 前半区间有多少空位
91     if(num<=k) return findd(o<<1,l,M,L,R,num); //如果num<=k 在前半找第num个
92     return findd(o<<1|1,M+1,r,L,R,num-k); // 在后半找第num - k 个
93 }
94 int main()
95 {
96     int _;
97     Cin(_);
98     while(_-- )
99     {
100         Memset(lazy,-1);
101         Cin(n);Cin(m);
102         build(1,1,n);

```

```
103     while(m--)  
104     {  
105         Cin(cmd);Cin(x);Cin(y);  
106         if(cmd == 1)  
107         {  
108             x++;  
109             int k = query(1,1,n,x,n); //判断从x位置到末尾还有多少空位  
110             if(k == 0)  
111                 puts("Can not put any one.");  
112             else{  
113                 int l = findd(1,1,n,x,n,1);  
114                 int r = findd(1,1,n,x,n,min(y,k));  
115                 Cout(l-1," ");  
116                 Cout(r-1,"\\n");  
117                 update(1,1,n,l,r); // 1到r置满  
118             }  
119         }  
120         if(cmd == 2)  
121         {  
122             x++;y++;  
123             int k = update2(1,1,n,x,min(n,y)); //x到y（或末尾）置空  
124             Cout(k,"\\n");  
125         }  
126     }  
127     puts("");  
128 }  
129 }
```