

信奥算法

数据结构

树状数组

BIT(Binary Indexed Tree)

二进制索引树

Fenwick Tree

经典问题

Range Sum Query

动态

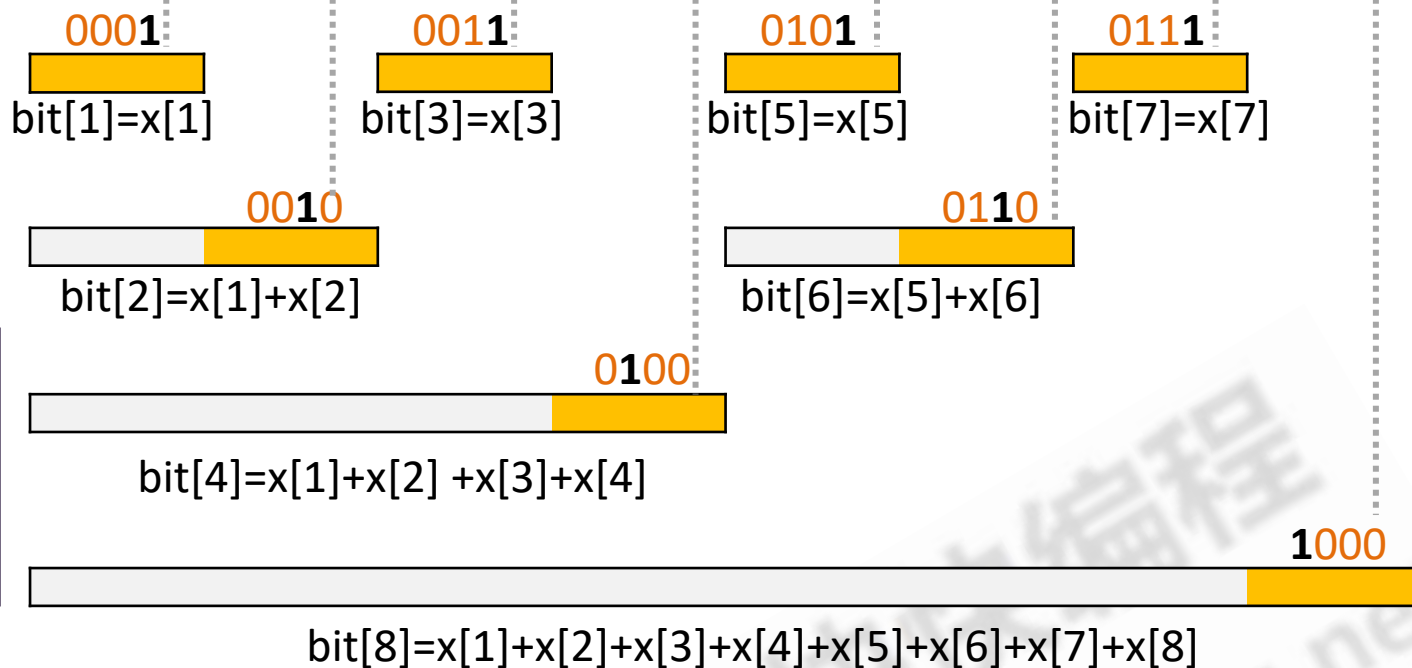
连续和查询

前缀和查询

Prefix Sum Query

BIT区间长度和端点

bit[i]	0	5	2	4	8	-1	-2	2	14
数值x[i]	空	5	-3	4	2	-1	-1	2	6
下标i	0	1	2	3	4	5	6	7	8



bit[i]对应区间
长度为二进制
末尾1对应的数
记作LSB(i)

Least
Significant
Bit

记
笔
记

bit[i] 记录 x数组的[i-LSB(i)+1,i]
编号区间内数值总和

核心代码

```
11 int LSB(int i){  
12     return i & (-i);  
13 }
```

原x数组i号元素加1时
更新bit数组

```
14 void add(int i){  
15     for(; i <= n; )  
16         bit[i]++;  
17 }
```

```
18 int psq(int i){  
19     int sum = 0;  
20     for(; i; i -= LSB(i))  
21           
22     return sum;  
23 }
```

prefix sum query



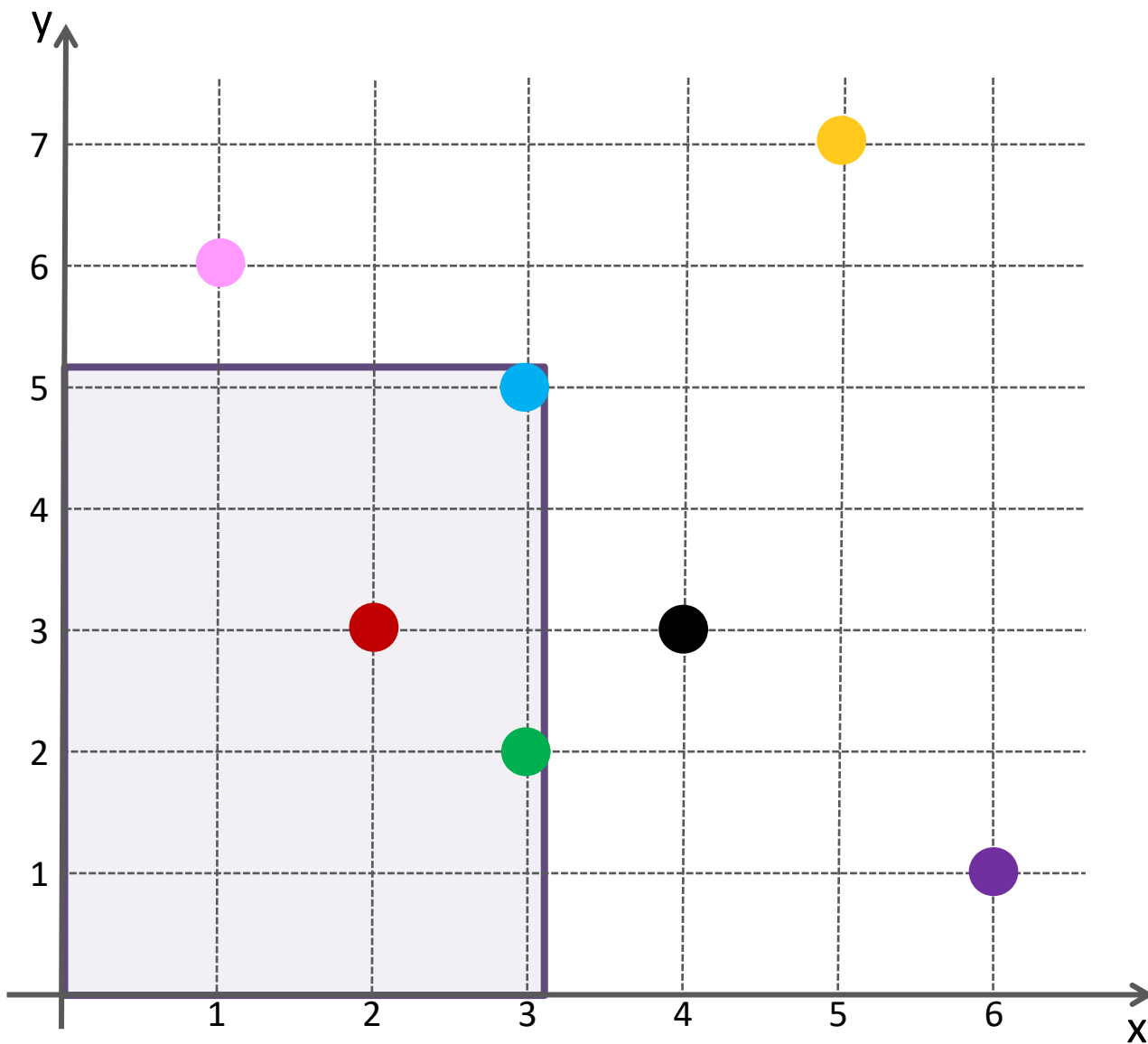
平面点左下角计数

n 个不同的平面点
对每个点统计共有几个其他点
横纵坐标都小于等于它

快快编程
kkcoding.net

左下角计数

横坐标小于等于当前点
纵坐标小于等于当前点



动态化：增加顺序概念

自然
顺序

输入先后顺序

题意中"时间"先后顺序

增加
顺序

按照数据大小先后处理

自定义排序后形成先后

动态化思想：增加"顺序"概念
让数据有"先后"地出现

这题用动态化思想
形成什么算法？

我想动动
别问我
动动是谁



动态

按照x坐标从小到大排序

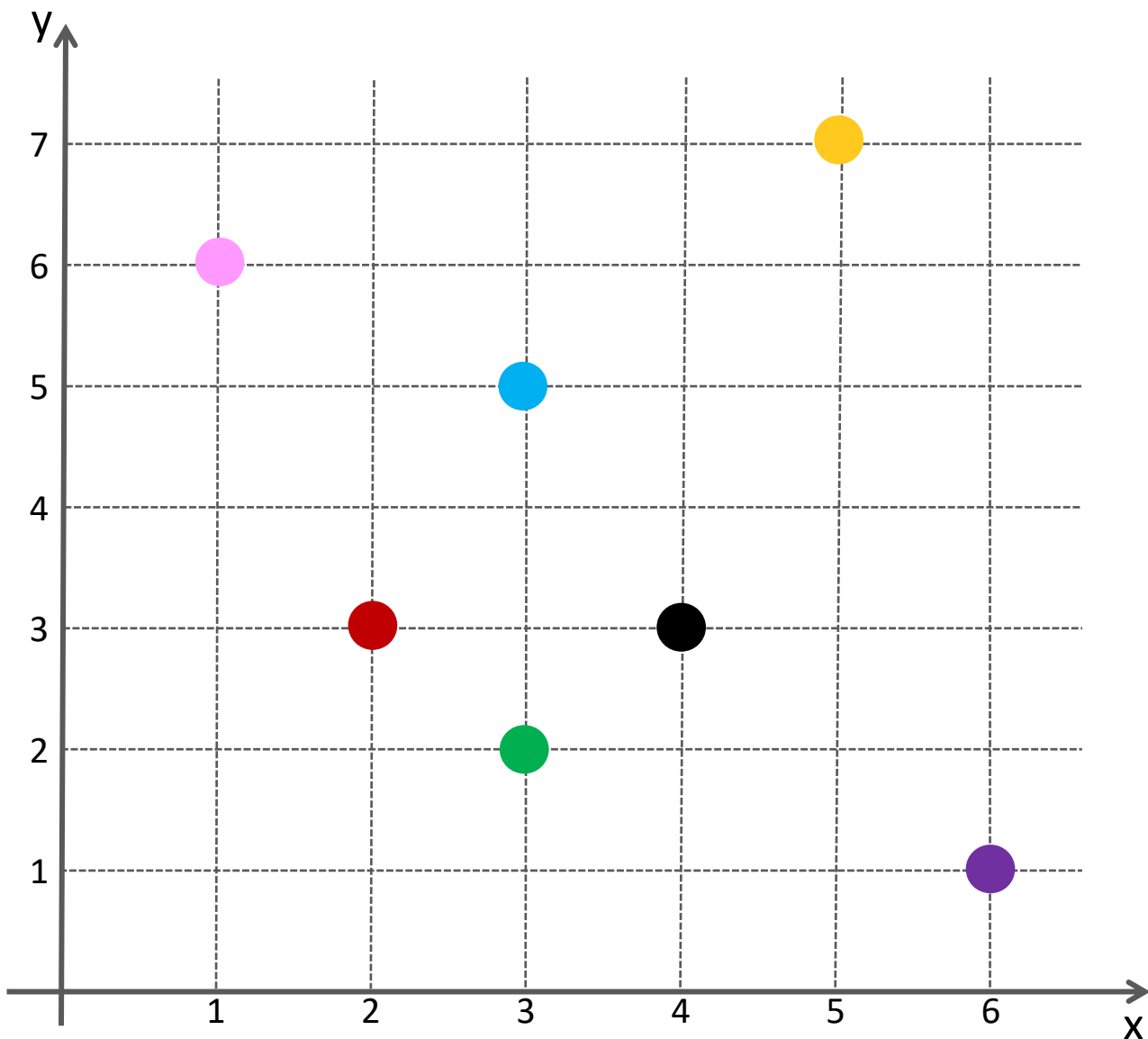
左下角计数

横坐标小于等于当前点
纵坐标小于等于当前点

已出现点里有几个点
y坐标小于等于当前点

y坐标计数器前缀和
用BIT数组拼凑

查询复杂度 $O(n\log Y)$

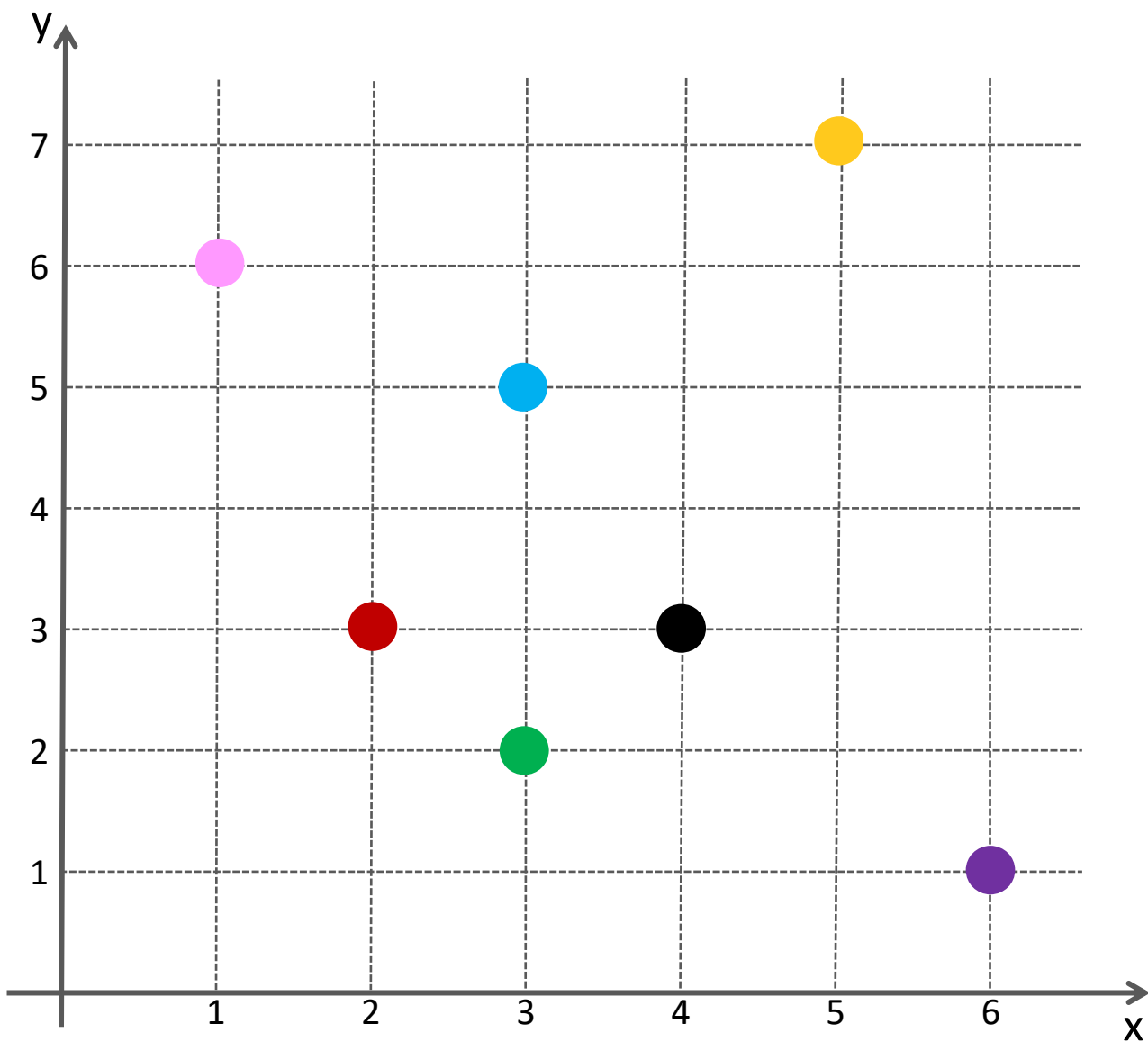



```
7 struct Point{int x,y,id;} p[N];
8 bool cmpX(const Point&a,const Point&b){
9     return a.x<b.x||a.x==b.x&&a.y<b.y;
10 }
```

```
27 cin>>n;
28 for(int i=1;i<=n;i++){
29     cin>>p[i].x>>p[i].y;
30     p[i].id=i;
31 }
32 sort(p+1,p+1+n,cmpX);
```

```
33 for(int i=1;i<=n;i++){
34     ans[p[i].id]=psq(p[i].y);
35     add(p[i].y);
36 }
37 for(int i=1;i<=n;i++)cout<<ans[i]<<" ";
```

动态 按照y坐标从小到大排序



左下角计数
横坐标小于等于当前点
纵坐标小于等于当前点

已出现点里有几个点
x坐标小于等于当前点

x坐标计数器前缀和
用BIT数组拼凑

查询复杂度 $O(n\log X)$

```
7 struct Point{int x,y,id;} p[N];
8 bool cmpY(const Point&a,const Point&b){
9     return a.y<b.y||a.y==b.y&&a.x<b.x;
10 }
```

```
27 cin>>n;
28 for(int i=1;i<=n;i++){
29     cin>>p[i].x>>p[i].y;
30     p[i].id=i;
31 }
32 sort(p+1,p+1+n,cmpY);
33 for(int i=1;i<=n;i++){
34     ans[p[i].id]=
35     }
36 }
37 for(int i=1;i<=n;i++)cout<<ans[i]<<" ";
```



动态化两种顺序如何选择

如果x坐标的范围大于y坐标的范围

按照x坐标从小到大排序
y坐标作为BIT数组编号

如果y坐标的范围大于x坐标的范围

按照y坐标从小到大排序
x坐标作为BIT数组编号



左上角计数

右上角计数

左下角计数

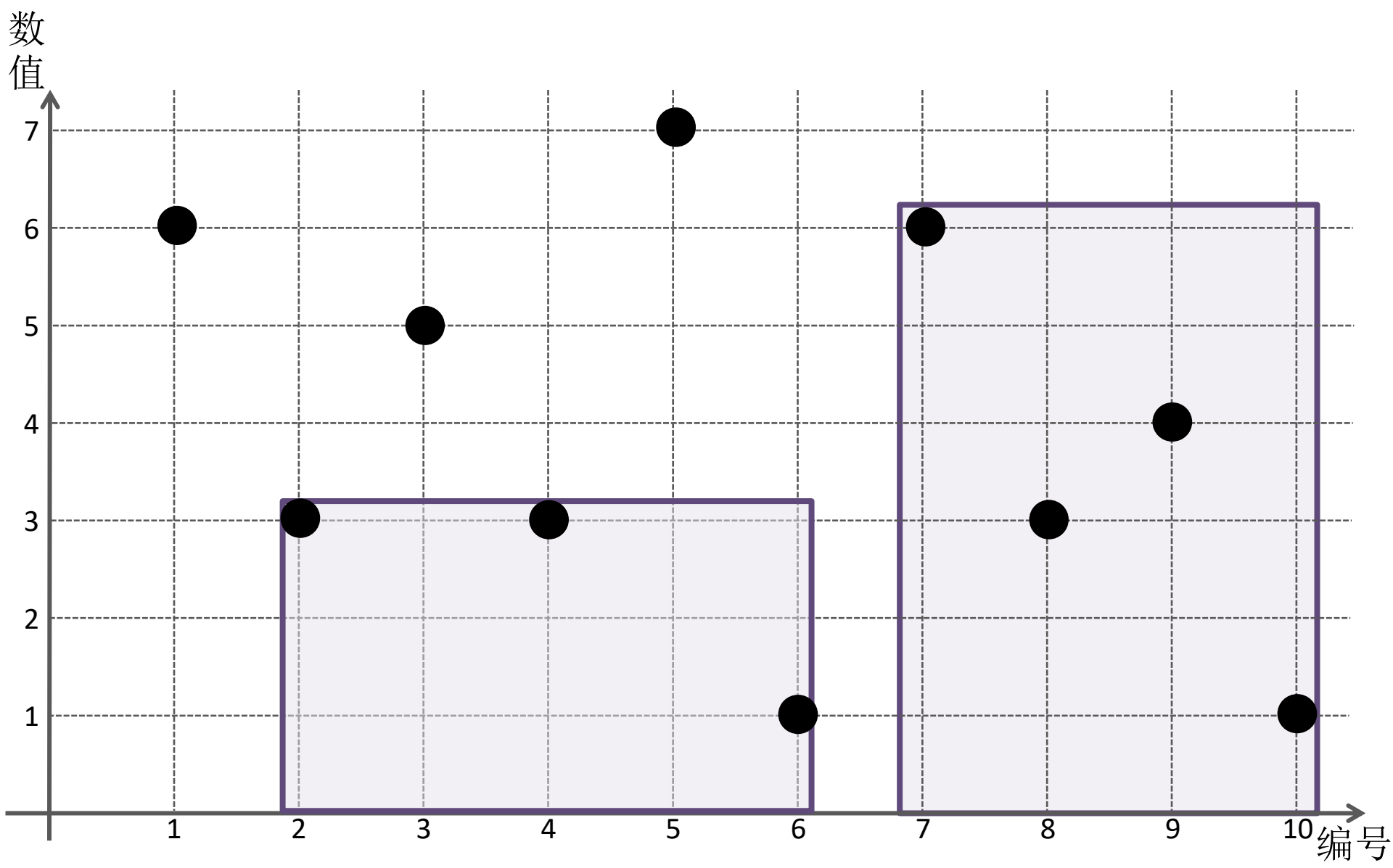
右下角计数



现场挑战 快快编程1058

快快编程
kkcoding.net

6	3	5	3	7	1	6	3	4	1
---	---	---	---	---	---	---	---	---	---





如果 y 坐标的范围大于 x 坐标的范围

按照 y 坐标从小到大排序
 x 坐标作为BIT数组编号

离线
询问

重新调整问题顺序
也按照 y 坐标从小到大排序

快快编程
kkcoding.net


```
8 struct Point{int x,y;} p[N];
9 bool cmpP(const Point&a,const Point&b){
10     return a.y<b.y||a.y==b.y&&a.x<b.x;
11 }
```

```
32 cin>>n>>m;
33 for(int i=1;i<=n;i++){
34     p[i].x=i;
35     cin>>p[i].y;
36 }
37 sort(p+1,p+1+n,cmpP);
```

```

12 struct Query{int l,r,k,id;} q[M];
13 bool cmpQ(const Query&a,const Query&b){
14     
15 }
    
```

```

38 for(int i=1;i<=m;i++){
39     cin>>q[i].l>>q[i].r>>q[i].k;
40     q[i].id=i;
41 }
42 sort(q+1,q+1+m,cmpQ);
    
```

双游标

```
43 int iP=1;
44 int iQ=1;
45 while(iQ<=m){
46     while(iP<=n&& p[iP].y<=q[iQ].k)
47         add(p[iP++].x);
48     ans[q[iQ].id]=
49     iQ++;
50 }
```

平面点依次出现

问询依次出现

以iQ号问询
的y坐标为界
相关平面点
逐个出现



现场挑战 快快编程1010

快快编程
kkcoding.net

动态

序列修改

第一步

定位第k位

第二步

移除第k位

第三步

尾部增加元素

请写出算法步骤或者难点

快编程
kkcoding.net

输入样例
5 4
2 3 5 3

数组模拟序列修改的弱点

删除第k位后,大量元素要整体平移

尝试

删除第k位后,该位置空着
不做任何平移

	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9	i=10
x[i]=	1	2	3	4	5	0	0	0	0	0
x[i]=	1	0	3	4	5	2	0	0	0	0
x[i]=	1	0	3	0	5	2	4	0	0	0
x[i]=	1	0	3	0	5	2	0	4	0	0
x[i]=	1	0	3	0	0	2	0	4	5	0

思考

如何定位第k个元素位置

二分法+BIT

search(2)

找到从左数第2个人的位置编号

psq(5)为2

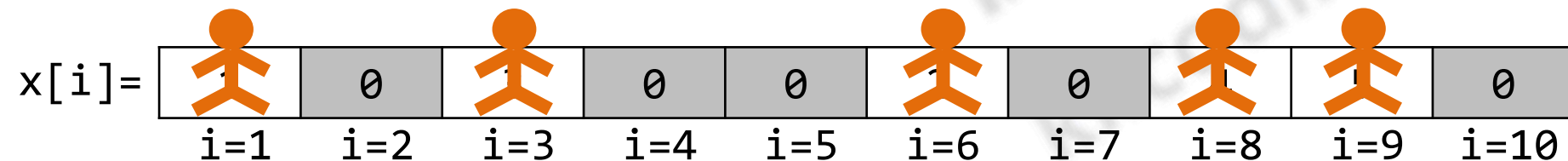
左侧5个位置里共2个人

psq(2)为1

左侧2个位置里共1个人

psq(3)为2

左侧3个位置里共2个人



思考

如何定位第k个元素位置

二分法+BIT

search(3)

找到从左数第3个人的位置编号

psq(5)为2

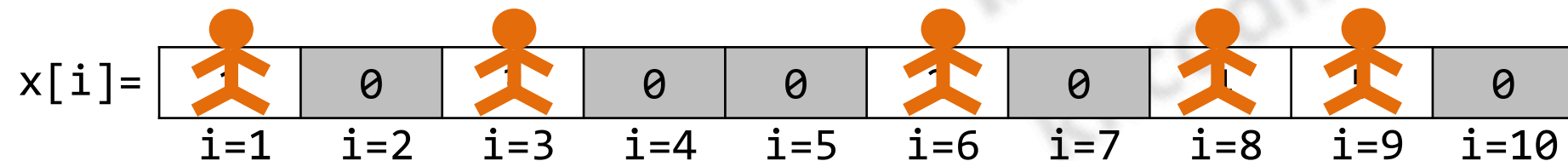
左侧5个位置里共2个人

psq(7)为3

左侧7个位置里共3个人

psq(6)为3

左侧6个位置里共3个人



最小化ans:
使得1号到ans号位置里至少有k个元素

```
18 int search(int k){  
19     int l=1,r=n+m,ans=n+m; ←  
20     while(l<=r){  
21         int mid=l+(r-l)/2;  
22         if(psq(mid)>=k) ←  
23             ans=mid,r=mid-1;  
24         else  
25             l=mid+1;  
26     }  
27     return ans;  
28 }
```

```
32      cin>>n>>m;
33      for(int i=1;i<=n;i++){
34          x[i]=i;
35          add(i,1);
36      }
37      for(int i=1;i<=m;i++){
38          int y;
39          cin>>y;
40          int pos=search(y);
41          x[n+i]=x[pos];
42          x[pos]=0;
43          add();
44          add();
45      }
46      cout<<x[n+m]<<endl;
```

动态

序列修改

第一步

定位第k位

第二步

移除第k位

第三步

尾部增加元素

特殊性

此题无需追踪所有人,只要关注最后1人

逆向思维/正难则反

利用时间倒流推理出信息

```
8      cin>>n>>m;
9      for(int i=1;i<=m;i++)cin>>x[i];
10     int rank=x[m];
11     for(int i=m-1;i>=1;i--){
12         if(rank==n)
13             
14         else if(x[i]<=rank)
15             
16     }
17     cout<<rank<<endl;
```

快快编程作业

2522

1058

1010

拓展题

989

请练习BIT解法