

太戈编程
etiger.vip

信奥算法

位运算

最低位置1代表的数值

Least **S**ignificant **B**it

```
int LSB(int a){  
    return a & (-a);  
}
```

表示a的二进制表示中最低位的1代表是十进制的几

十进制	二进制
0	00000000
a	10110100
0 - a	01001100
a & (-a)	00000100

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

快快编程698

读题后请同学简述题意

点更新

段查询

动态问题

整段总和查询

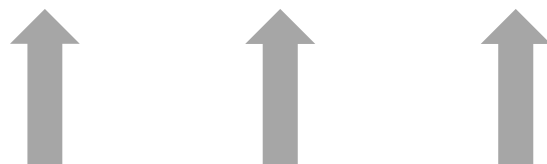
Range Sum Query
简称RSQ



动态RSQ 连续和查询

方法	别名	预计算/ 初始化	单次查 询	点更新	代码长 度	常数
在线枚举	暴力	-	$O(N)$	$O(1)$	短	小
前缀和 数组	Prefix sums	$O(N)$	$O(1)$	$O(N)$	短	小
分块	根号 算法	$O(N)$	$O(\sqrt{N})$	$O(1)$	中	小
线段树	ST树	$O(N)$	$O(\log N)$	$O(\log N)$	长	大
树状数组	BIT	$O(N)$	$O(\log N)$	$O(\log N)$	短	小

能得
几分



平衡各种操作的复杂度

为什么

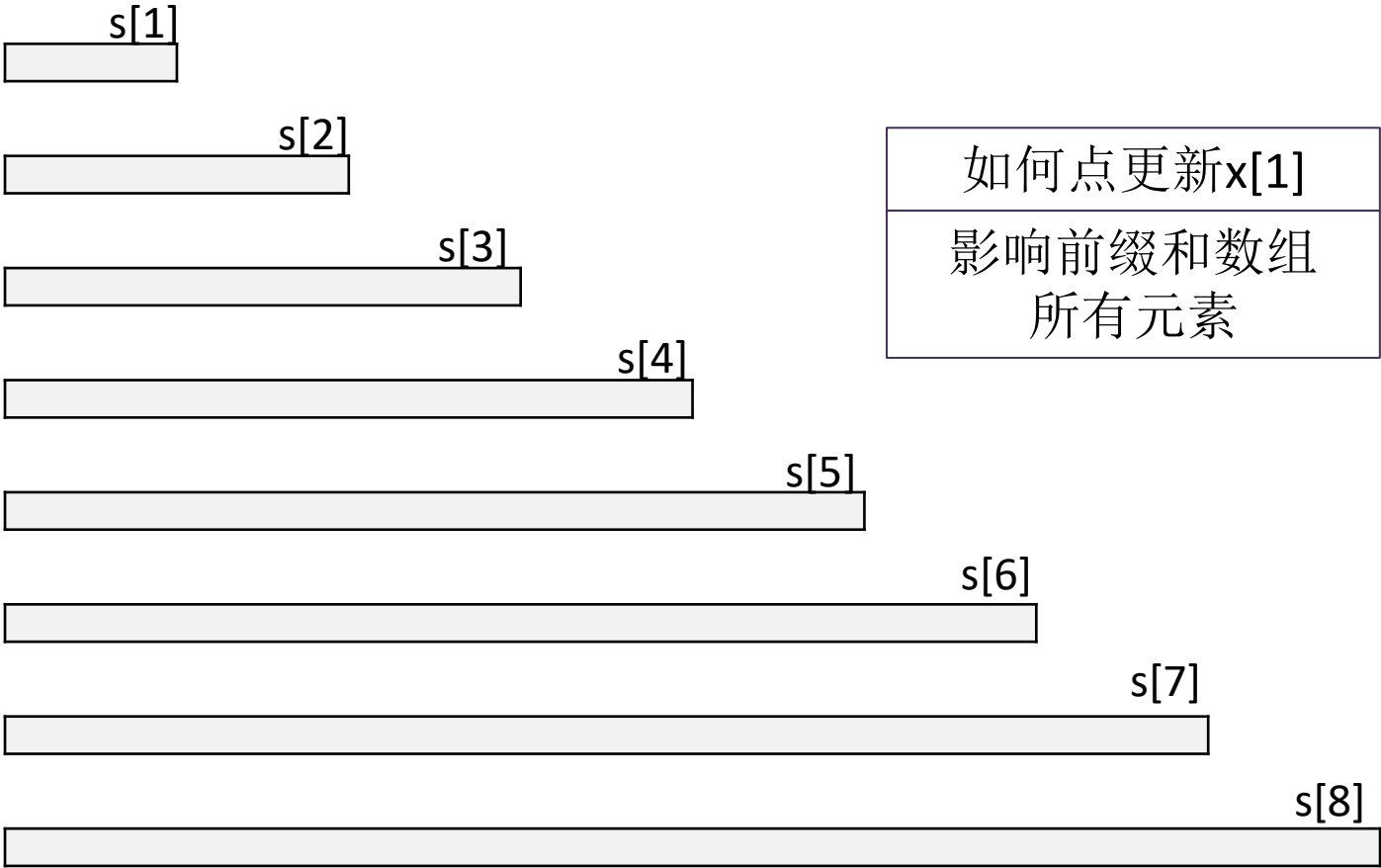
记录图中区间总和

前缀和数组

前缀和s[i]	0	5	2	6	8	7	6	8	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

查询飞快
点更新太慢

如何点更新x[1]
影响前缀和数组
所有元素



线段树

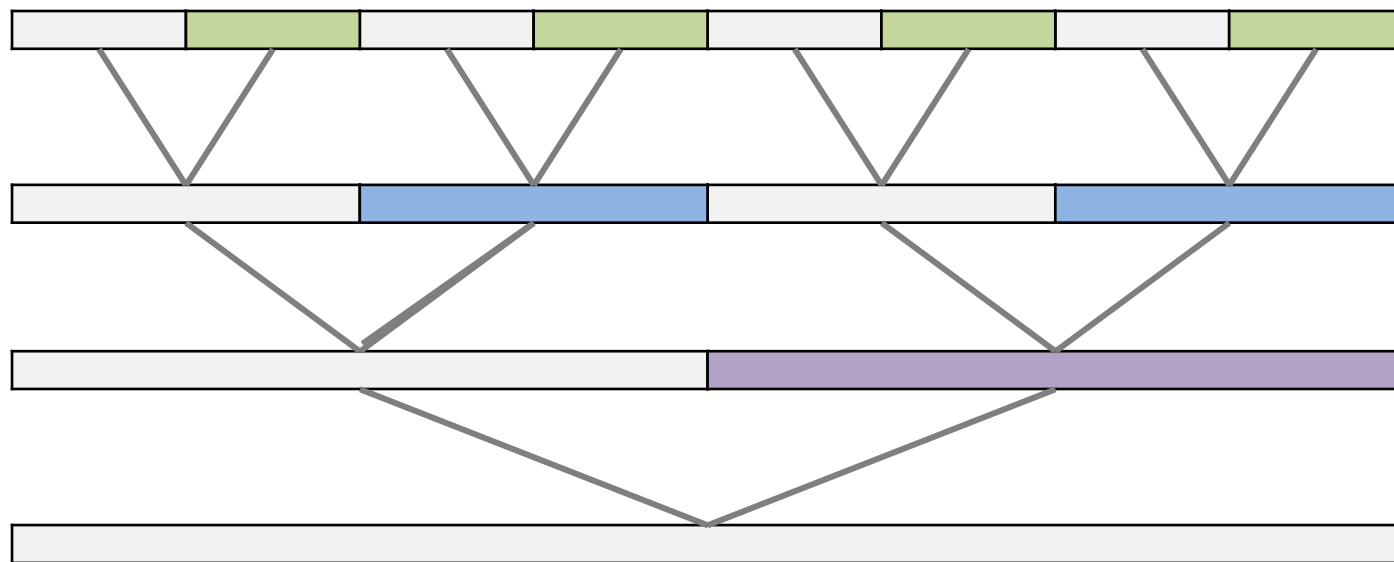
前缀和s[i]	0	5	2	6	8	7	6	8	14
数值x[i]	空	5	-3	4	2	-1	-1	2	6
下标i	0	1	2	3	4	5	6	7	8

查询 $O(\log N)$

点更新 $O(\log N)$

查询[1,7]时
哪些区间贡献

点更新1号时
哪些区间更新

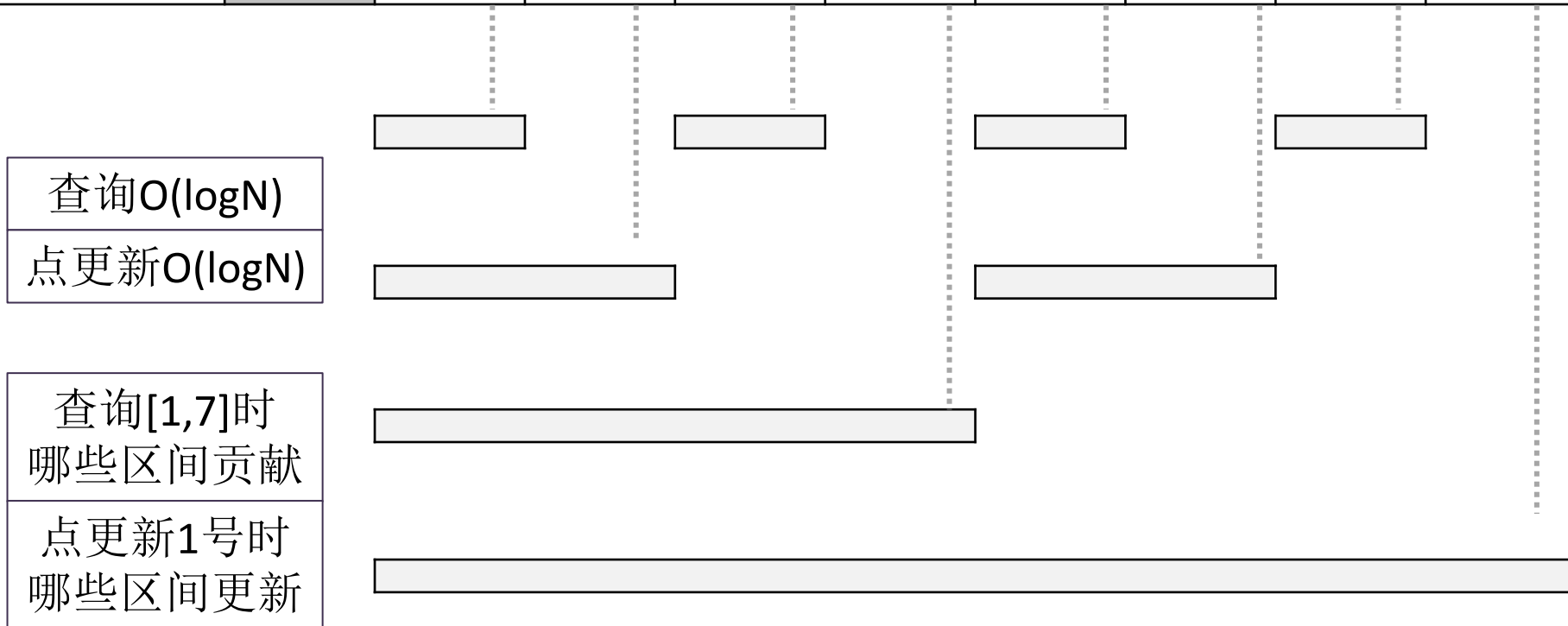


所有前缀区间都可用左儿子们拼接

查询前缀和时
右儿子们都多余

树状数组(BIT)

前缀和s[i]	0	5	2	6	8	7	6	8	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



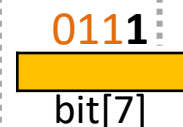
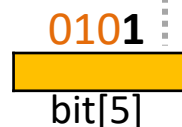
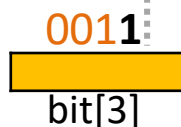
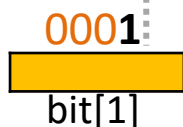
恰好剩n个区间
该如何编号

思考：查询后缀和
该如何处理

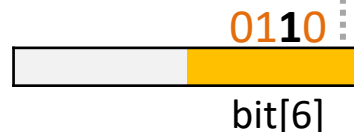
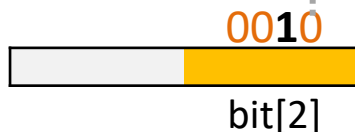
树状数组(BIT) 编号

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

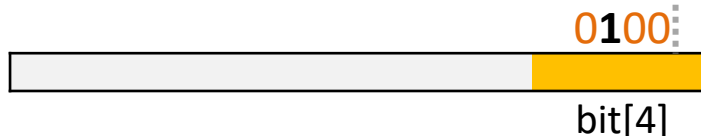
二进制 **???1**
区间长度 1



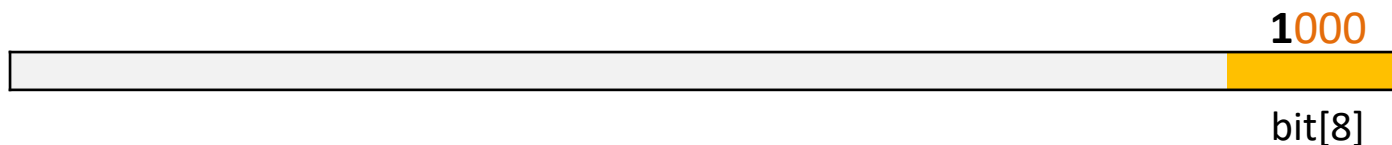
二进制 **??10**
区间长度 2



二进制 **?100**
区间长度 4



二进制 **1000**
区间长度 8



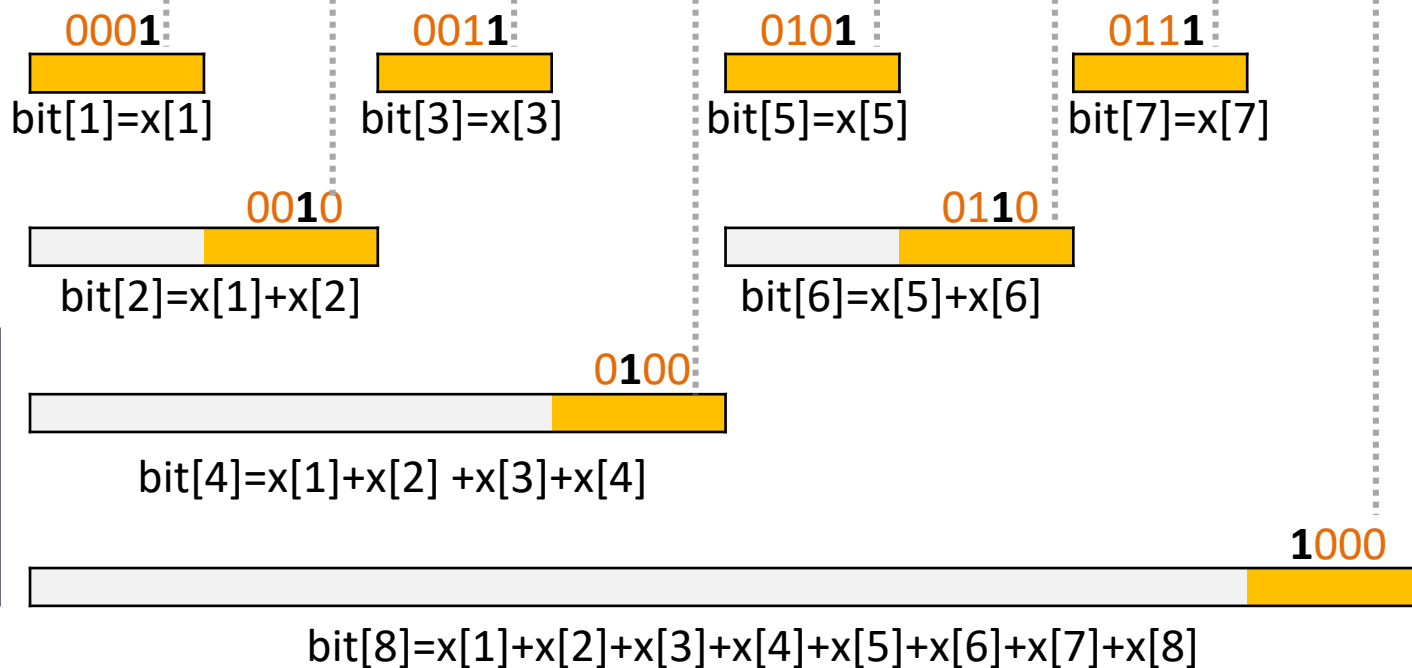
恰好剩n个区间
该如何编号

区间右端点
作为编号

区间长度为二进制
末尾1对应的数

树状数组(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)

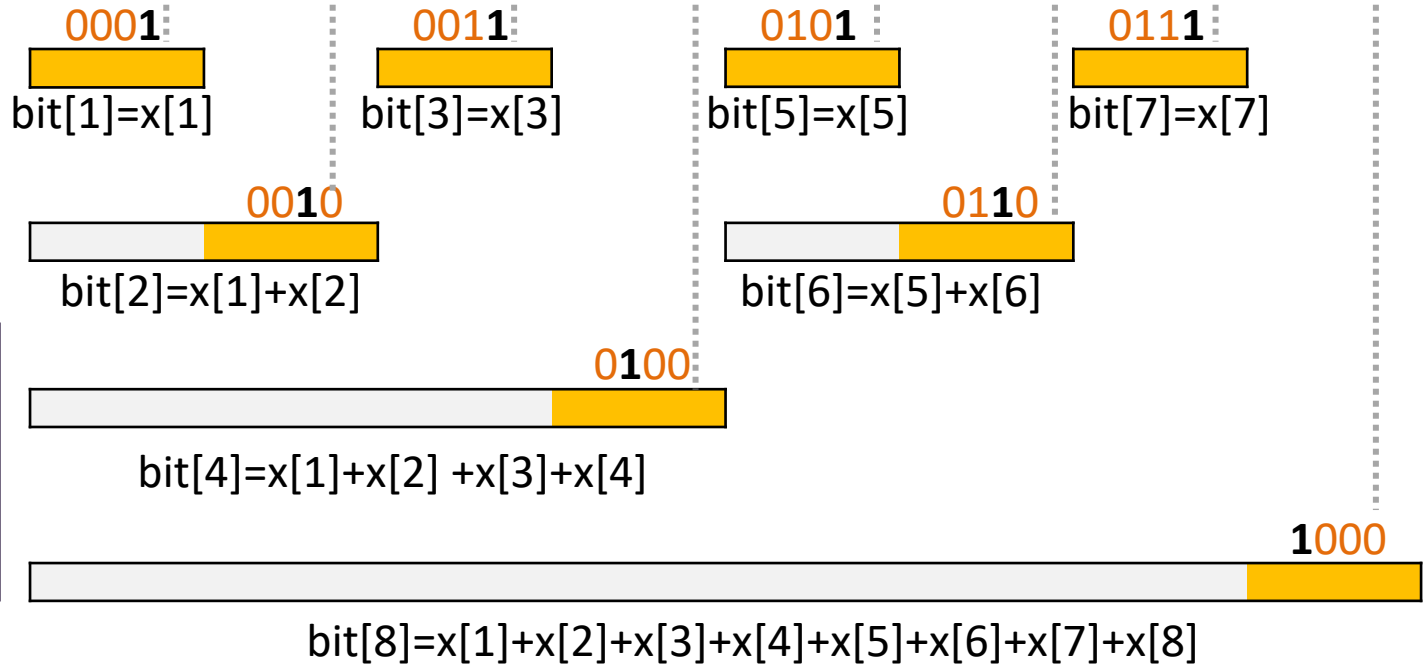
Least
Significant
Bit

bit[i]对应区间左端点
 $i - \text{LSB}(i) + 1$

bit[i]对应区间右端点
 i

树状数组(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

LSB(i)

i号区间长度为LSB(i)	别称 lowbit(i)
---------------	--------------

二进制 **???1**
区间长度 1

二进制 **??10**
区间长度 2

二进制 **?100**
区间长度 4

二进制 **1000**
区间长度 8

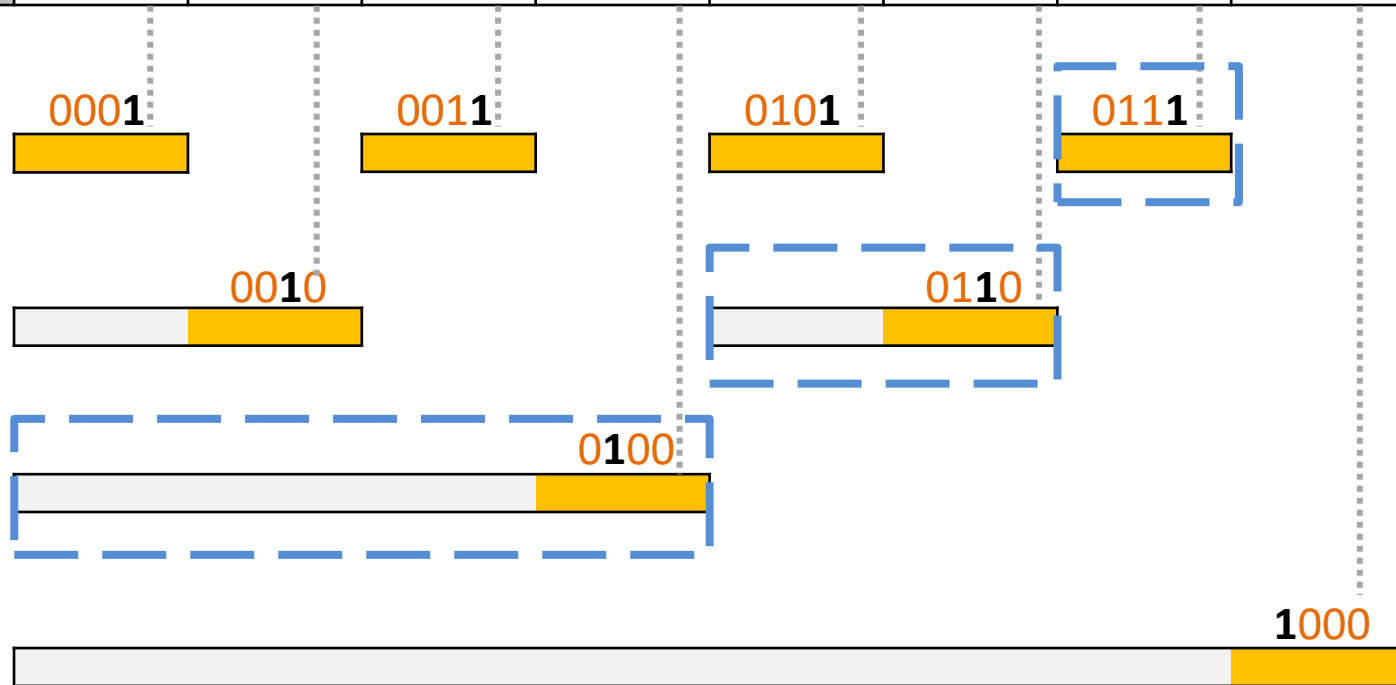
十进制	二进制
0	00000000
x	10110 1 00
0 - x	01001 1 00
$x \& (-x)$	00000 1 00

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

查询[1,7]前缀和

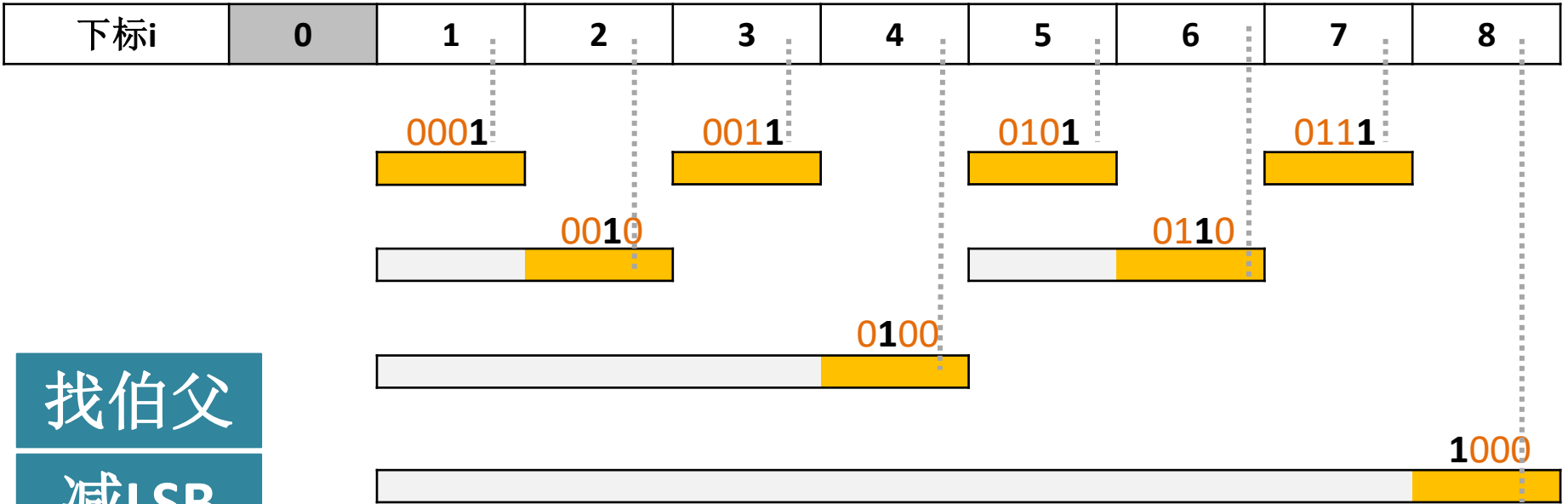
前缀和s[i]	0	5	2	6	8	7	6	8	14
数值x[i]	空	5	-3	4	2	-1	-1	2	6
下标i	0	1	2	3	4	5	6	7	8

找伯父
减LSB



查询[1,7]前缀和	[1,7]拆成[1,4]+[5,6]+[7,7]
$7_{10} = 0111_2$ 对应 0100 ₂ 0110 ₂ 0111 ₂	

前缀和查询代码



找伯父

减LSB

prefix
sum
query

```
12 11 psq(11 i) {  
13     int sum=0;  
14     while(i)  
15         sum=(sum+bit[i])%MOD, i-=LSB(i);  
16     return sum;  
17 }
```

返回[1,i]的前缀和

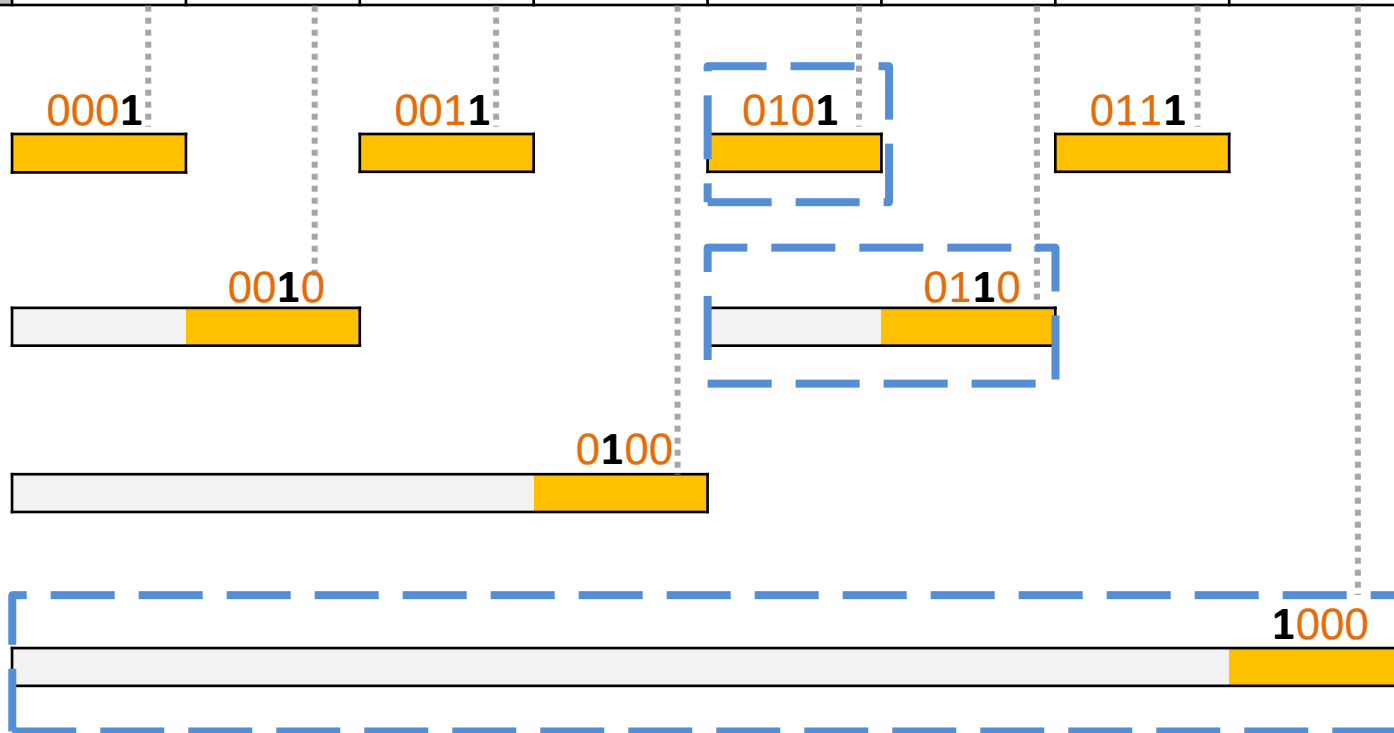
点更新5号

前缀和s[i]	0	5	2	6	8	7	6	8	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

找爸爸
或爸爸的爸

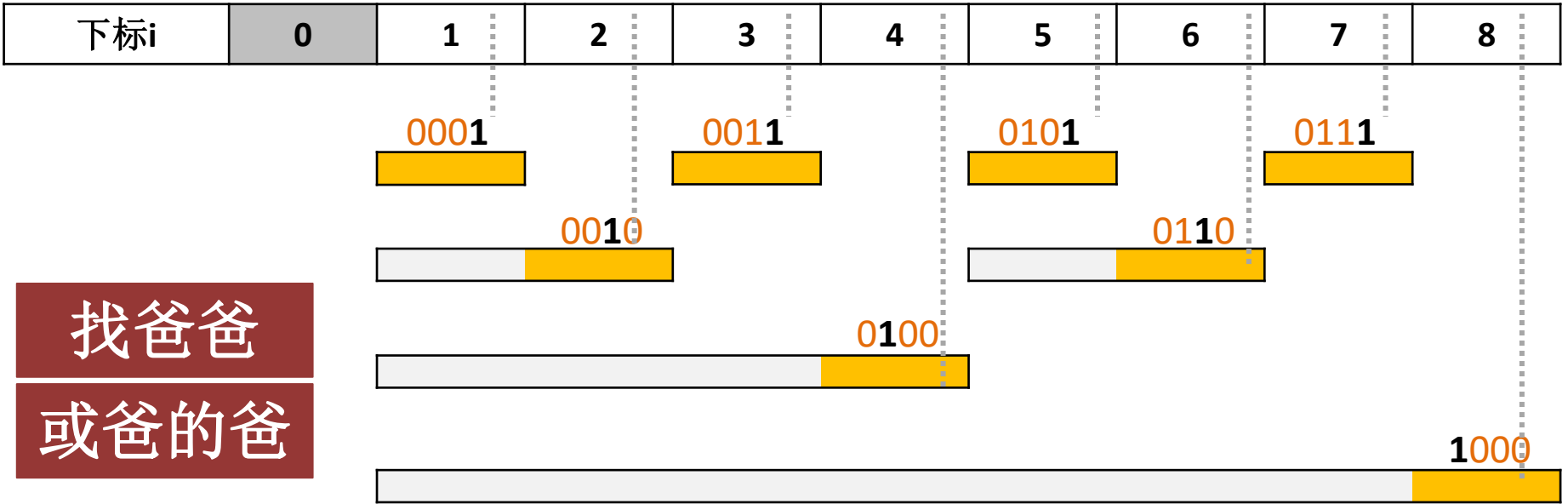
.....

加LSB



点更新5号	修改[5,5],[5,6],[1,8]
右端点的二进制数 0101_2 0110_2 1000_2	

点更新代码



找爸爸
或爸爸的爸

.....

加LSB

将i号数值增加z

```
8 void add(ll i, ll z) {  
9     while(i <= n)  
10         bit[i] += z, i += LSB(i);  
11 }
```

树状数组(BIT) - 核心代码

```
11 LSB(11 i){return ;}
```

```
8 ☐ void add(11 i, 11 z) {  
9     while()  
10         bit[i] += z, ;  
11 }
```

```
12 ☐ 11 psq(11 i) {  
13     int sum = 0;  
14     while()  
15         sum = (sum + bit[i]) % MOD, ;  
16     return sum;  
17 }
```

计数器数组 连续和/前缀和

BIT应用

如何处理
多个问询

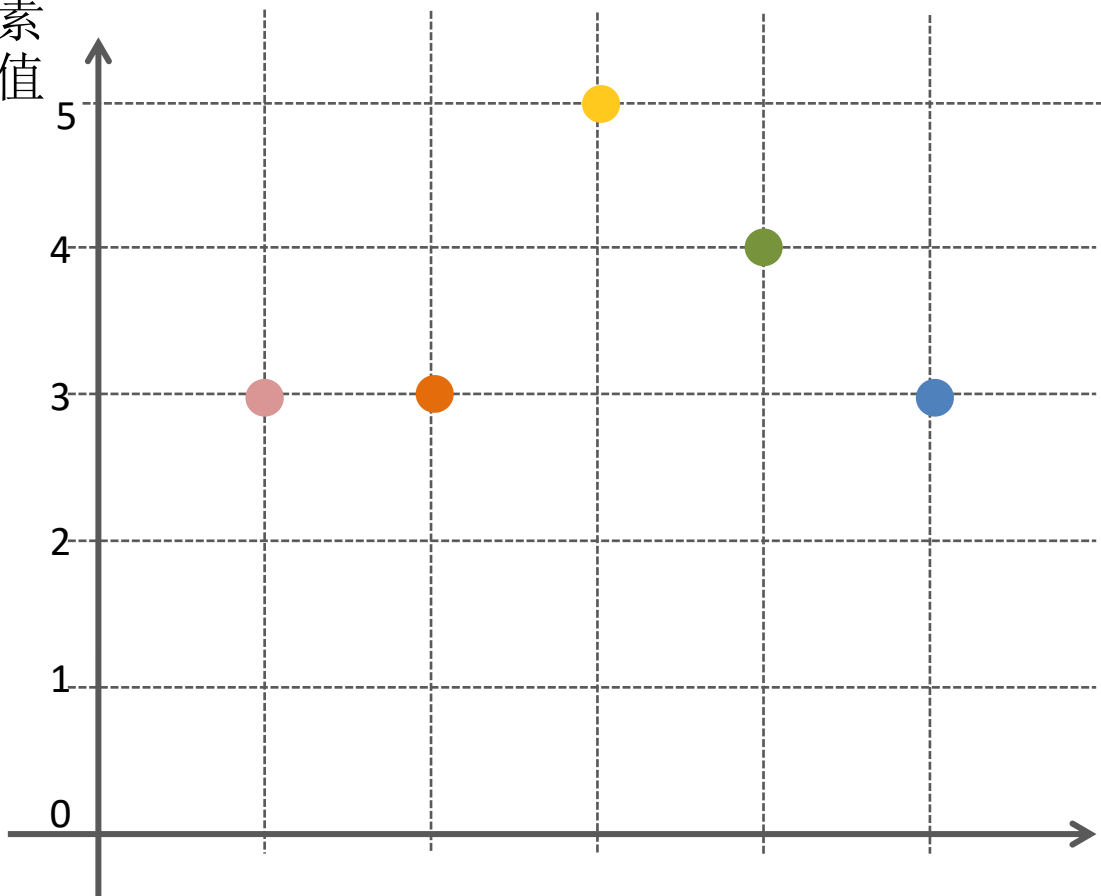
有几个数小于等于3?
有几个数小于等于4?

当时计数器数组
对应的前缀和

原始
数组x

数值	3	3	5	4	3
下标	x[1]	x[2]	x[3]	x[4]	x[5]

元素
数值



c[5] 1

c[4] 1

c[3] 3

c[2] 0

c[1] 0

c[0] 0

数组
下标

瞬时排名

今年的宇宙编程大赛共 n 位选手入围决赛，全是Lester徒弟。只有一次网络提交机会， n 位选手依次提交，第 i 个提交的选手显示出得分为 $x[i]$ ，即刻他的分数就会记录到得分排行榜上，对于每个选手请你求出他提交的时刻是**当时第几名**?名次可并列。

输入一行为正整数 n ， $2 \leq n \leq 200000$ 。接着一行为 n 个非负整数，第 i 个数为 $x[i]$ ，均不超过10000。←

输出一行共 n 个整数，由空格隔开。

输入样例

4
0 2 4 3

输出样例

1 1 1 2

输入样例

5
59 59 61 60 59

输出样例

1 1 1 2 3

得分 a 的排名=比 a 分数高的人数+1

=总人数-不超过 a 分数的人数+1

计数问题

动态

模拟：依次分析每个选手 i
询问当时有几个分数比 $x[i]$ 大

计数器前缀和数组

计数器数组也有对应的前缀和数组，下标对应值域

计数器 数组c	计数	0	0	0	3	1	1	0	0
	下标	c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]

计数器 前缀和 数组sc	计数	0	0	0	3	4	5	5	5
	下标	sc[0]	sc[1]	sc[2]	sc[3]	sc[4]	sc[5]	sc[6]	sc[7]

若原数组新增一个
数值为0的数

计数器 数组c	计数	1	0	0	3	1	1	0	0
	下标	c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]

计数器 前缀和 数组sc	计数	1	1	1	4	5	6	6	6
	下标	sc[0]	sc[1]	sc[2]	sc[3]	sc[4]	sc[5]	sc[6]	sc[7]

计数器前缀和数组

前缀和数组编号必须从1号开始
不可以从0号开始

为了让计数器数组对应的
前缀和数组编号从1号开始
数值值域要平移到最小值为1

值域为-100到100的整数时如何处理？

每个数加101，值域平移到1到201

计数器树状数组(cBIT)

计数器数组有对应的前缀和数组

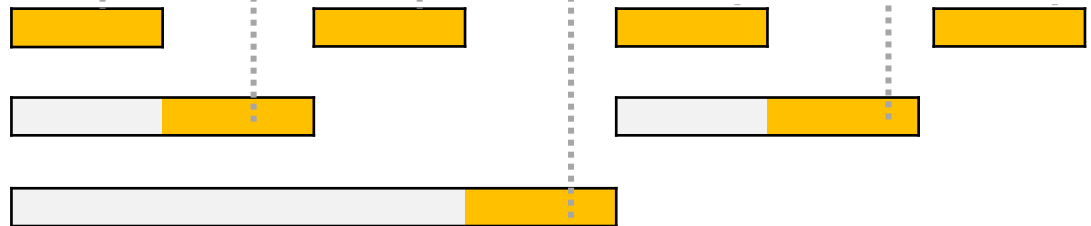
计数器数组也有对应的BIT

计数器数组c	计数	0	0	0	3	1	1	0	0
	下标	c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]

值域映射到1开头

新计数器数组c'	计数	0	0	0	0	3	1	1	0
	下标	c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]

计数器
树状数
组cbit



$cbit[i]$ 记录 c 数组的 $[i-LSB(i)+1, i]$ 编号范围里数值总和

计数器树状数组(cBIT)

计数器数组有对应的前缀和数组

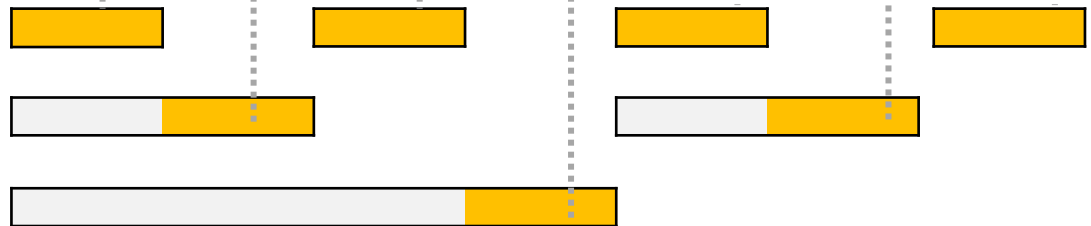
计数器数组也有对应的BIT

计数器数组c	计数	0	0	0	3	1	1	0	0
	下标	c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]

值域映射到1开头

新计数器数组c'	计数	0	0	0	0	3	1	1	0
	下标	c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]

计数器
树状数
组cbit



$cbit[i]$ 记录 x 数组数值为 $[i-LSB(i)+1, i]$ 范围内的有几个

瞬时排名

n 位选手依次提交，第 i 个提交的选手显示出得分为 $x[i]$ ，求出他提交的时刻是**当时第几名**？名次可并列。

输入样例

4

2 0 4 3

输出样例

1 2 1 2

动态
维护
 $cbit[]$

为了控制
最低分为1分
所有人加1分

提交2分	变成3分	修改 $cbit[3], cbit[4]$				
$cbit[k]$	空	0	0	1	1	0
下标k	0	1	2	3	4	5

此时有几人分数小于等于3分？

共 $cbit[3] + cbit[2] = 1$ 人

此时已提交总人数1人

所以此人排名 $1 - 1 + 1 = 1$

提交0分	变成1分	$cbit[1], cbit[2], cbit[4]$				
$cbit[k]$	空	1	1	1	2	0
下标k	0	1	2	3	4	5

瞬时排名

n 位选手依次提交，第 i 个提交的选手显示出得分为 $x[i]$ ，求出他提交的时刻是**当时第几名**？名次可并列。

输入样例

4

2 0 4 3

输出样例

1 2 1 2

动态
维护

cbit[]

为了控制
最低分为1分
所有人加1分

20

21

22

23

24

25

26

```
cin>>n;
for (ll i=1;i<=n;i++) {
    cin>>x;
    x++;
    add(x,1);
    cout<<i-psq(x)+1<<" ";
}
```

太戈编程

698

843

拓展题

844,122

作业要求

需要用BIT表完成