

太戈编程
etiger.vip

信奥算法

连续段简称"段"
也称作"区间"

更新操作

点更新

单点修改

单点增加

段更新

整段修改

整段增加

段更新比点更新更难
因为段包含点

查询操作

单点查询

段查询

整段求和查询

整段最值查询

段查询比点查询更难
因为段包含点

太戈编程698

读题后请同学简述题意

点更新

段查询

动态
问题

单点增加

整段求和查询

Range Sum Query
简称RSQ

封装隔离：更新+查询

```
15 int main(){
16     cin>>n>>m;
17     for(ll i=1;i<=m;i++){
18         ll t,x,y;
19         cin>>t>>x>>y;
20         if(t==2) add(x,y);
21         else cout<<rsq(x,y)<<endl;
22     }
23     return 0;
24 }
```

单点增加

将x号数字增加y

整段求和查询

从x号到y号求和

算法1

暴力累加解决RSQ

算法2

前缀和做差解决RSQ

暴力枚举

```
7 void add(ll x, ll y){  
8     a[x]=(a[x]+y)%MOD;  
9 }
```

单点增加

将x号数字增加y

```
10 ll rsq(ll l, ll r){  
11     ll sum=0;  
12     for(int i=l; i<=r; i++)  
13         sum=(sum+a[i])%MOD;  
14     return sum;  
15 }
```

整段求和查询

从l号到r号求和

易错点

要求取模

前缀和

```
7 void add(ll x, ll y){  
8     a[x]=(a[x]+y)%MOD;  
9     for(ll i=x;i<=n;i++)  
10         s[i]=(s[i-1]+a[i])%MOD;  
11 }
```

单点增加

将x号数字增加y

可以改 $s[i]=(s[i]+y)\%MOD$

```
12 ll rsq(ll l, ll r){  
13     ll sum=s[r]-s[l-1];  
14     sum=(sum%MOD+MOD)%MOD;  
15     return sum;  
16 }
```

整段求和查询

从1号到r号求和

易错点

负数取模

时间复杂度

方法	别名	点更新	单次查询	总时间复杂度
在线枚举	暴力	$O(1)$	$O(N)$	$O(MN)$
前缀和	Prefix Sums	$O(N)$	$O(1)$	$O(MN)$

能得
几分

请同学总结2种算法
优点和缺点

能否结合两者优点?

分块算法

$$O(\sqrt{N})$$

大卸八块

编号i	1	2	3	4	5	6	7	8	9	10
a[i]	3	1	4	1	5	9	2	6	5	3

编号i	1	2	3	4	5	6	7	8	9	10
b[i]	1	1	1	2	2	2	3	3	3	4

第1块	第2块	第3块	第4块
-----	-----	-----	-----

段查询[3,10]区间内数字总和, 怎么做?

区间拼凑 $[3,3] + [4,6] + [7,9] + [10,10]$

编号i	1	2	3	4
bs[i]	8	15	13	3

bs[k]代表k号块的块内总和

编号i	1	2	3	4	5	6	7	8	9	10
a[i]	3	1	4	1	5	19	2	6	5	3

编号i	1	2	3	4	5	6	7	8	9	10
b[i]	1	1	1	2	2	2	3	3	3	4

第1块	第2块	第3块	第4块
-----	-----	-----	-----

点更新：6号数字增加10

维护块总和

更新变为两步

编号i	1	2	3	4
bs[i]	8	25	13	3

bs[k]代表k号块的块内总和

数组含义

a数组

$a[i]$ 代表原数组*i*号数值

共*n*个

b数组

$b[i]$ 代表原*i*号元素对应块号

共*n*个

bs数组

$bs[k]$ 代表第*k*块的块内总和

有几块?

共有 $b[n]$ 块

单点增加

将x号数字增加y

```
7 void add(ll x, ll y){  
8     a[x]=(a[x]+y)%MOD;  
9     bs[b[x]]=(bs[b[x]]+y)%MOD;  
10 }
```

更新原数组a[]

更新分块和数组bs[]

整段求和查询

从1号到r号求和

```
11 rsq(l, r){
12     s=0;
13     if(b[l]==b[r]){
14         for(int i=l;i<=r;i++)s=(s+a[i])%MOD;
15     }else{
16         for(int i=l;b[i]==b[l];i++)s=(s+a[i])%MOD;
17         for(int i=r;b[i]==b[r];i--)s=(s+a[i])%MOD;
18         for(int k=b[l]+1;k<=b[r]-1;k++)s=(s+bs[k])%MOD;
19     }
20     return s;
21 }
```

高频易错点

区间首尾在同一块里

分类
讨论

左右端点在同一块

左右端点不在同一块

块的大小

编号i	1	2	3	4	5	6	7	8	9	10
b[i]	1	1	1	2	2	2	3	3	3	4

设每块块内放L个数

L取值为几, 算法最快?

每块大小L越大

块的数量越小

$L=n$, 共1块

退化成纯"暴力"

每块大小L越小

块的数量越大

$L=1$, 共n块

退化成纯"暴力"

块的大小

编号i	1	2	3	4	5	6	7	8	9	10
b[i]	1	1	1	2	2	2	3	3	3	4

设每块块内放L个数


L取值为几, 算法最快?

$$\text{块大小 } L * \text{块数} = n$$

$$\text{块数} = n/L$$

$$\text{取 } L=\sqrt{N}, \text{块数}=\sqrt{N}$$

```
23 | cin>>n>>m;  
24 | L=sqrt(n);  
25 | for(ll i=1;i<=n;i++)b[i]=(i-1)/L+1;
```



时间复杂度

方法	别名	点更新	单次查询	总时间复杂度
在线枚举	暴力	$O(1)$	$O(N)$	$O(MN)$
前缀和	Prefix Sums	$O(N)$	$O(1)$	$O(MN)$
分块	根号算法	$O(1)$	$O(\sqrt{N})$	$O(M\sqrt{N})$

对于 $O(M\sqrt{N})$ 复杂度的算法
请估计 M, N 多大规模会超时?

分块算法小结

2层暴力：整块暴力+块内暴力

区间拼凑

头块+尾块+中间块

维护块内总和

信息整体打包

易错点

区间首尾两端点在同一块内

太戈编程676

读题后请同学简述题意

无更新

段查询

静态问题

整段最值查询

Range Maximum Query
简称RMQ

编号i	1	2	3	4	5	6	7	8	9	10
a[i]	3	1	4	1	5	9	2	6	5	3

编号i	1	2	3	4	5	6	7	8	9	10
b[i]	1	1	1	2	2	2	3	3	3	4

第1块	第2块	第3块	第4块
-----	-----	-----	-----

段查询[3,10]区间内最大值, 怎么做?

区间拼凑 $[3,3] + [4,6] + [7,9] + [10,10]$

编号i	1	2	3	4
bm[i]	4	9	6	3

bm[k]代表k号块的块内最大值

```

17 int main(){
18     cin>>n>>m;
19     for(ll i=1;i<=n;i++)cin>>a[i];
20     L=sqrt(n);
21     for(ll i=1;i<=n;i++)b[i]=
22     for(ll k=1;k<=b[n];k++){
23         ll r=
24         ll l=
25         bm[k]=*max_element(a+l,a+r+1);
26     }
27     for(ll i=1;i<=m;i++){
28         ll l,r;
29         cin>>l>>r;
30         cout<<rmq(l,r)<<endl;
31     }
32     return 0;
33 }

```

补全
程序

整段最值查询

l号到r号最大值

```
6 11 rmq(11 l, 11 r){
7     11 ans=a[l];
8     if(b[l]==b[r])
9         ans=*max_element(a+l, a+r+1);
10    else{
11        for(11 i=l; b[i]==b[l]; i++) ans=max(ans, a[i]);
12        for(11 i=r; b[i]==b[r]; i--) ans=max(ans, a[i]);
13        for(11 k=b[l]+1; k<=b[r]-1; k++) ans=max(ans, bm[k]);
14    }
15    return ans;
16 }
```

高频易错点

区间首尾在同一块里

左右端点在同一块

左右端点不在同一块

偶像天团

西佳佳偶像天团共 k 人，最近 n 年每年有一位歌手加入。当人数超过 k 人时老团员自动退团。第 i 人的颜值为 $x[i]$ ，团内颜值最低者成为团长，求 k 人成团后每年的团长颜值是多少。

输入第一行为 n 和 k ，第二行为 n 个正整数。 $1 \leq k \leq n \leq 200000$ ，每个数都不超过100000。输出共 $n-k+1$ 个数字，由空格隔开。

输入样例： 输出样例：

5 3

7 7 6

7 8 9 7 6

固定长度滑动RMQ

`rmq()`函数改成`min`

你有没有爱上我



偶像天团

```
17 int main(){
18     cin>>n>>k;
19     for(int i=1;i<=n;i++)cin>>a[i];
20     L=sqrt(n);
21     for(int i=1;i<=n;i++)b[i]=
22     for(int j=1;j<=b[n];j++){
23         int r=j*L;
24         int l=r-L+1;
25         bm[j]=
26     }
27     for(int i=k;i<=n;i++)
28         cout<<rmq( )<<" ";
29     return 0;
30 }
```

补全
程序

太戈编程1713

读题后请同学简述题意

单点更新

段查询

动态
问题

单点修改

整段最值查询

Range Maximum Query
简称RMQ

编号i	1	2	3	4	5	6	7	8	9	10
a[i]	3	1	4	1	5	4	2	6	5	3

编号i	1	2	3	4	5	6	7	8	9	10
b[i]	1	1	1	2	2	2	3	3	3	4

第1块	第2块	第3块	第4块
-----	-----	-----	-----

点更新：6号数字改为4

维护块最值

更新变为两步

编号i	1	2	3	4
bm[i]	4		6	3

bm[k]代表k号块的块内最大值

```

6 void update(ll x, ll y){
7     a[x]=y;
8     ll r=b[x]*L;
9     ll l=r-L+1;
10    bm[b[x]]=*max_element(a+l, a+r+1);
11 }

```

单点修改

将x号数字改为y

能否加速?

更新原数组a[]

更新分块最值数组bm[]

重新块内打擂台

单点修改

将x号数字改为y

```
6 void update(ll x, ll y){
7     if(y < a[x] && a[x] == bm[b[x]]){
8         a[x] = y;
9         ll r = b[x] * L;
10        ll l = r - L + 1;
11        bm[b[x]] = *max_element(a + l, a + r + 1);
12    }
13    else {
14        a[x] = y;
15        bm[b[x]] = max(bm[b[x]], y);
16    }
17 }
```

块内老擂主变小

块内老擂主不变小

块内不用重新打擂台
单挑老擂主即可

太戈编程667

读题后请同学简述题意

段更新

点查询

动态
问题

整段增加

查询单个数值

编号i	1	2	3	4	5	6	7	8	9	10
a[i]	3	1	14	1	5	9	12	16	5	3

编号i	1	2	3	4	5	6	7	8	9	10
b[i]	1	1	1	2	2	2	3	3	3	4

第1块	第2块	第3块	第4块
-----	-----	-----	-----

段更新：3号到8号数字都增加10

懒标记Lazy-tag

整块更新先拖延

编号i	1	2	3	4
c[i]	0	10	0	0

c[k]代表k号块内整体被延后的增量

尚未更新

编号i	1	2	3	4	5	6	7	8	9	10
a[i]	3	1	14	1	5	9	12	16	5	3

编号i	1	2	3	4	5	6	7	8	9	10
b[i]	1	1	1	2	2	2	3	3	3	4

第1块	第2块	第3块	第4块
-----	-----	-----	-----

点查询：6号数值是几？

懒标记Lazy-tag

整块更新先拖延

编号i	1	2	3	4
c[i]	0	10	0	0

c[k]代表k号块内整体被延后的增量

尚未更新

数组含义

a数组

$a[i]$ 代表原数组*i*号数值

b数组

$b[i]$ 代表原*i*号元素对应块号

c数组

$c[k]$ 代表*k*号块内整体被延后的增量

易错点

$a[i]$ 不再代表当前*i*号真实数值
请问当前*i*号真实数值如何表示?


```

16 int main(){
17     cin>>n>>m;
18     L=sqrt(n);
19     for(ll i=1;i<=n;i++)b[i]=(i-1)/L+1;
20     for(ll i=1;i<=m;i++){
21         ll t,x,y,z;
22         cin>>t>>x>>y>>z;
23         if(t==1) add(x,y,z);
24         else cout<< <<endl;
25     }
26     return 0;
27 }

```

补全
程序

请同学实现`add(l, r, z)`函数
段更新：将`l`号到`r`号增加`z`

```
7 void add(ll l, ll r, ll z){
8     if(b[l]==b[r])
9         for(ll i=l; i<=r; i++) a[i]=(a[i]+z)%MOD;
10    else{
11        for(ll i=l; b[i]==b[l]; i++) a[i]=(a[i]+z)%MOD;
12        for(ll i=r; b[i]==b[r]; i--) a[i]=(a[i]+z)%MOD;
13        for(ll k=b[l]+1; k<=b[r]-1; k++) c[k]=(c[k]+z)%MOD;
14    }
15 }
```

分块算法小结

区间拼凑

头块+尾块+中间块

懒标记**Lazy-tag**

整块更新先拖延

易错点

头尾端点在同一块内

易错点

元素真实数值尚未更新

如何查错？

打印所有数组a[],b[],bm[],bs[]等
比对手算结果

易错点

头尾端点在同一块内

易错点

元素真实数值尚未更新

拓展任务

有同学认为 $L=\sqrt{n}$
并不是最优的块大小

能否找到让程序更快的块大小 L

太戈编程

676

1713

667

拓展题

698, 657, 663, 961