

树状数组

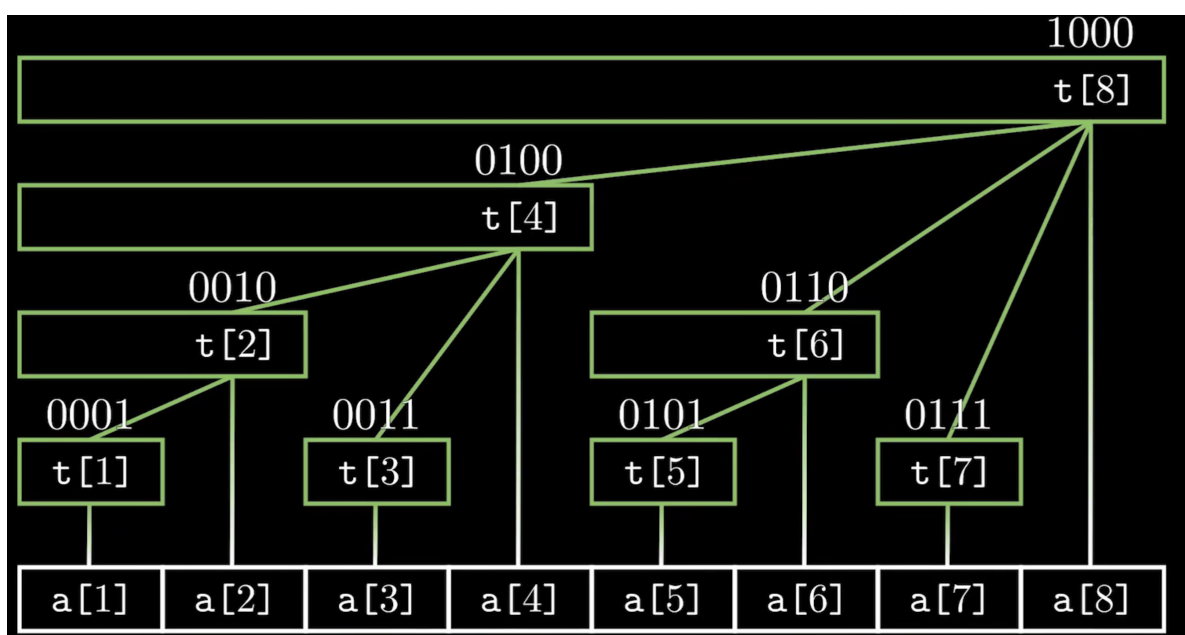
算法作用

- 动态维护前缀和、异或和、最大值、最小值
- 单点修改 | 查询前缀和、单点修改 | 单点查询、单点修改 | 区间查询、区间修改 | 单点查询
- 无法维护最大值

前置知识

$\text{lowbit}(x)$ 运算: $x \& -x$

一些公式

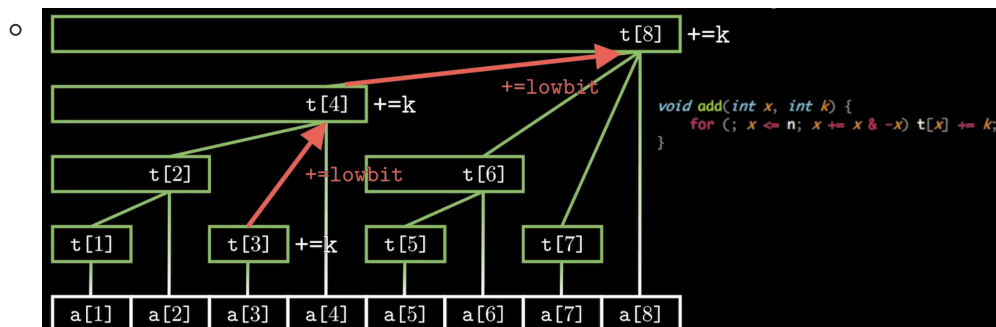


对于任意一个数 x :

- $t[x]$ 保存以 x 为根的子树中叶节点的和.
- $t[x]$ 节点所覆盖的长度等于 $\text{lowbit}(x)$
- $t[x]$ 节点的父节点为 $t[x + \text{lowbit}(x)]$
- $t[x]$ 节点的左侧相邻节点为 $t[x - \text{lowbit}(x)]$
- 整个树的深度是 $\log_2(n + 1)$

操作

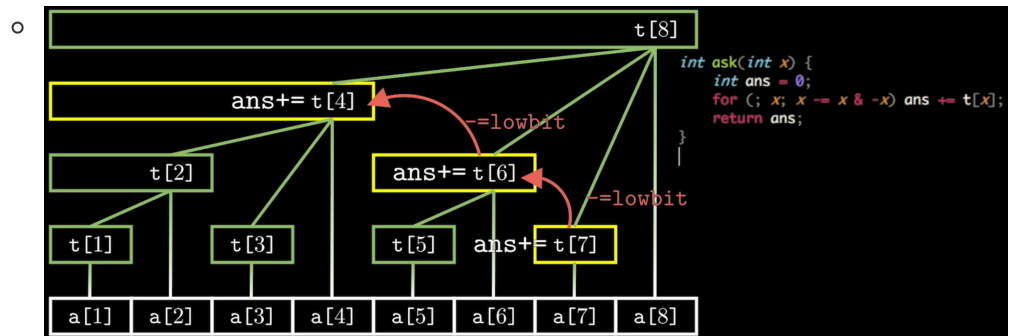
- $\text{add}(x, k)$ — 对 $a[x]$ 加上一个数 k



- o 依次向上修改父节点($idx += \text{lowbit}(idx)$), 并对所有父节点进行更新

```
1 void add(int x, int k){
2     for(; x <= n; x += lowbit(x)) t[x] += k;
3 }
```

- **ask(x)** — 查询从 $a[1]$ 到 $a[x]$ 区间的前缀和



- o 从该点向左上找到与其相邻的节点($idx -= \text{lowbit}(idx)$)

```
1 int ans(int x){
2     int ans = 0;
3     for(; x; x -= lowbit(x)) ans += t[x];
4     return ans;
5 }
```

- 单点修改 | 查询前缀和

```
1 add(x, k);
2 ask(k);
```

- 单点修改 | 单点查询

```
1 add(x, k);
2 ask(x) - ask(x - 1);
```

- 单点修改 | 区间查询

```
1 add(x, k);
2 ask(r) - ask(l - 1);
```

- 区间修改 | 单点查询 **差分**

注意: 需要用树状数组维护原数组的差分数组的前缀和

```
1 add(l, d), add(r + 1, -d);
2 a[x] = ask(x);
```

- 区间修改 | 区间查询

① 变差为 b_i^2

$a[l \sim r], \text{ 求 } b[l] + \dots + b[r]$ $b[l] + \dots + b[r] = C$

② $a_1 + \dots + a_x$

$$= \sum_{i=1}^x a_i = \sum_{i=1}^x \sum_{j=1}^i b_j$$

1
2
3

	b_1	b_2	...	b_x	
1	b_1	b_2	...	b_x	
2	b_1	b_2	b_3	...	b_x
3	b_1	b_2	b_3	...	b_x
...					
x	b_1	b_2	b_3	...	b_x

$$(b_1 + b_2 + \dots + b_x) \times (x+1) - (b_1 - 2b_2 - 3b_3 - \dots - xb_x)$$

\downarrow
 $i \cdot b_i$ 的前缀和

tr1. b_i 的前缀和

tr2. $i \cdot b_i$ 的前缀和

```

1 //初始化
2 add(tr1, i, b);
3 add(tr2, i, i * b);
4 //添加
5 add(l, d, tr1), add(r + 1, -d, tr1);
6 add(l, l * d, tr2), add(r + 1, (r + 1) * -d, tr2);
7 //查询
8 int get(int x){
9     return ask(x, tr1) * (x + 1) - ask(x, tr2);
10 }
11 get(r) - get(l - 1)

```

例题

P3374 【模板】树状数组 1

```

1 #include <bits/stdc++.h>
2 #define endl "\n"
3 #define int long long
4
5 using namespace std;
6
7 const int maxn = 5e5 + 10;
8 int tr[maxn];
9 int a[maxn];
10 int n, m;
11
12 int lowbit(int x){
13     return x & -x;
14 }
15
16 void add(int x, int b){
17     for(; x <= n; x += lowbit(x)) tr[x] += b;
18 }
19
20 int ask(int x){

```

```

21     int ans = 0;
22     for(; x; x -= lowbit(x)) ans += tr[x];
23     return ans;
24 }
25
26 signed main(){
27     ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
28
29     cin >> n >> m;
30     for(int i = 1; i <= n; i++){
31         cin >> a[i];
32         add(i, a[i]);
33     }
34     while(m -- ){
35         int op; cin >> op;
36         if(op == 1){
37             int a, b; cin >> a >> b;
38             add(a, b);
39         }
40         else{
41             int a, b; cin >> a >> b;
42             cout << ask(b) - ask(a - 1) << endl;
43         }
44     }
45
46
47     return 0;
48 }

```

P3368 【模板】树状数组 2

```

1  #include <bits/stdc++.h>
2  #define endl "\n"
3  #define int long long
4  using namespace std;
5
6  const int maxn = 5e5 + 10;
7  int n, m;
8  int tr[maxn], a[maxn];
9
10 int lowbit(int x){
11     return x & -x;
12 }
13
14 void add(int x, int k){
15     for(; x <= n; x += lowbit(x)) tr[x] += k;
16 }
17
18 int ask(int x){
19     int ans = 0;
20     for(; x; x -= lowbit(x)) ans += tr[x];
21     return ans;
22 }
23
24 signed main(){
25     ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
26     cin >> n >> m;

```

```

27     for(int i = 1; i <= n; i++){
28         cin >> a[i];
29         add(i, a[i] - a[i - 1]);
30     }
31
32     while(m--){
33         int op; cin >> op;
34         if(op == 1){
35             int x, y, k; cin >> x >> y >> k;
36             add(x, k), add(y + 1, -k);
37         }
38         else{
39             int k; cin >> k;
40             cout << ask(k) << endl;
41         }
42     }
43
44
45     return 0;
46 }

```

AcWing 241. 楼兰图腾

```

1  #include <bits/stdc++.h>
2  #define endl "\n"
3  #define int long long
4  using namespace std;
5
6  const int maxn = 2e5 + 10;
7  int n, m;
8  int ans1, ans2;
9  int a[maxn], upper[maxn], lower[maxn];
10 int tr[maxn];
11
12 int lowbit(int x){
13     return x & -x;
14 }
15
16 void add(int a, int b){
17     for(; a <= n; a += lowbit(a)) tr[a] += b;
18 }
19
20 int ask(int x){
21     int ans = 0;
22     for(; x; x -= lowbit(x)) ans += tr[x];
23     return ans;
24 }
25
26 signed main(){
27     ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
28     cin >> n;
29     for(int i = 1; i <= n; i++) cin >> a[i];
30
31     for(int i = 1; i <= n; i++){
32         int y = a[i];
33         lower[i] = ask(y - 1);
34         upper[i] = ask(n) - ask(y);

```

```
35
36     add(y, 1);
37 }
38
39 memset(tr, 0, sizeof tr);
40
41 for(int i = n; i >= 1; i --){
42     int y = a[i];
43     ans2 += ask(y - 1) * lower[i];
44     ans1 += (ask(n) - ask(y)) * upper[i];
45     add(y, 1);
46 }
47
48 cout << ans1 << " " << ans2;
49
50 return 0;
51 }
```