

시그 트리와 팰린 트리

소스 코드

최백준 [choi@startlink.io](mailto:choi@startlink.io)

---

C++14

```
1 #include <stdio>
2 #include <cmath>
3 #include <cassert>
4 #include <vector>
5 #include <algorithm>
6 using namespace std;
7 int maximum(vector<int> &tree, int node, int start, int end, int left, int right) {
8     if (left > end || right < start) {
9         return 0;
10    }
11    if (left <= start && end <= right) {
12        return tree[node];
13    }
14    return max(maximum(tree, node*2, start, (start+end)/2, left, right), maximum(tree, node*2+1,
15    (start+end)/2+1, end, left, right));
16 }
17 void update(vector<int> &tree, int node, int start, int end, int i, int value) {
18     if (i > end || i < start) {
19         return;
20     }
21     tree[node] = max(tree[node], value);
22     if (start != end) {
23         update(tree, node*2, start, (start+end)/2, i, value);
24         update(tree, node*2+1, (start+end)/2+1, end, i, value);
25     }
26 }
27 const int MAX = 1000000;
28 int main() {
29     int n;
30     scanf("%d", &n);
31     int h = (int)ceil(log2(MAX));
32     int tree_size = (1 << (h+1));
33     vector<int> tree(tree_size);
34     int ans = 0;
35     for (int i=0; i<n; i++) {
36         int num;
37         scanf("%d", &num);
38         int cur = maximum(tree, 1, 1, MAX, 1, num-1);
39         if (ans < cur + 1) {
40             ans = cur + 1;
41         }
42         update(tree, 1, 1, MAX, num, cur+1);
43     }
44     printf("%d\n", ans);
45     return 0;
46 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

9372 KB

620 ms

1275 B

C++14

```
1 #include <cstdio>
2 #include <map>
3 #include <cmath>
4 #include <cassert>
5 #include <vector>
6 #include <algorithm>
7 using namespace std;
8 int maximum(vector<int> &tree, int node, int start, int end, int left, int right) {
9     if (left > end || right < start) {
10         return 0;
11     }
12     if (left <= start && end <= right) {
13         return tree[node];
14     }
15     return max(maximum(tree, node*2, start, (start+end)/2, left, right), maximum(tree, node*2+1,
16 (start+end)/2+1, end, left, right));
17 }
18 void update(vector<int> &tree, int node, int start, int end, int i, int value) {
19     if (i > end || i < start) {
20         return;
21     }
22     tree[node] = max(tree[node], value);
23     if (start != end) {
24         update(tree, node*2, start, (start+end)/2, i, value);
25         update(tree, node*2+1, (start+end)/2+1, end, i, value);
26     }
27 }
28 const int MAX = 1000000;
29 int main() {
30     int n;
31     scanf("%d", &n);
32     vector<int> a(n);
33     vector<int> b(n);
34     for (int i=0; i<n; i++) {
35         scanf("%d", &a[i]);
36         b[i] = a[i];
37     }
38     sort(b.begin(), b.end());
39     b.erase(unique(b.begin(), b.end()), b.end());
40     map<int, int> d;
41     for (int i=0; i<b.size(); i++) {
42         d[b[i]] = i+1;
43     }
44     for (int i=0; i<n; i++) {
45         a[i] = d[a[i]];
46     }
47     int h = (int)ceil(log2(MAX));
48     int tree_size = (1 << (h+1));
49     vector<int> tree(tree_size);
50     int ans = 0;
51     for (int i=0; i<n; i++) {
52         int num = a[i];
53         int cur = maximum(tree, 1, 1, MAX, 1, num-1);
54         if (ans < cur + 1) {
55             ans = cur + 1;
56         }
57         update(tree, 1, 1, MAX, num, cur+1);
58     }
59     printf("%d\n", ans);
60     return 0;
61 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

63644 KB

1896 ms

1627 B

C++14

중간  
시작

```
1 #include <cstdio>
2 #include <vector>
3 #include <algorithm>
4 #include <map>
5 using namespace std;
6 int l;
7 void update(vector<int> &tree, int i, int diff) {
8     while (i <= l) {
9         tree[i] += diff;
10        i += (i & -i);
11    }
12 }
13 int sum(vector<int> &tree, int i) {
14     int ans = 0;
15     while (i > 0) {
16         ans += tree[i];
17         i -= (i & -i);
18     }
19     return ans;
20 }
21 int sum(vector<int> &tree, int l, int r) {
22     if (l > r) return 0;
23     return sum(tree, r) - sum(tree, l-1);
24 }
25 int main() {
26     int n;
27     scanf("%d",&n);
28     vector<int> a(n);
29     for (int i=0; i<n; i++) {
30         scanf("%d",&a[i]);
31     }
32     map<int,int> position;
33     for (int i=0; i<n; i++) {
34         int num;
35         scanf("%d",&num);
36         position[num] = i+1;
37     }
38     long long ans = 0;
39     vector<int> tree(n+1);
40     l = n;
41     for (int i=0; i<n; i++) {
42         ans += (long long)sum(tree, position[a[i]]-1, n);
43         update(tree, position[a[i]], 1);
44     }
45     printf("%lld\n",ans);
46     return 0;
47 }
```

binary (tree)

$N \times O(\log N)$

$O(N \log N)$

결과

메모리

시간

코드 길이

맞았습니다!!

28508 KB

980 ms

985 B

C++14

```
1 #include <cstdio>
2 #include <vector>
3 using namespace std;
4 int l = 0;
5 void update(vector<int> &tree, int i, int diff) {
6     while (i <= l) {
7         tree[i] += diff;
8         i += (i & -i);
9     }
10 }
11 int sum(vector<int> &tree, int i) {
12     int ans = 0;
13     while (i > 0) {
14         ans += tree[i];
15         i -= (i & -i);
16     }
17     return ans;
18 }
19 int main() {
20     int t;
21     scanf("%d",&t);
22     while (t--) {
23         int n,m;
24         scanf("%d %d",&n,&m);
25         vector<int> tree(n+m+1);
26         vector<int> position(n+1);
27         l = n+m;
28         for (int i=1; i<=n; i++) {
29             update(tree, m+i, 1);
30             position[i] = m+i;
31         }
32         ~ for (int i=m; i>=1; i--) {
33             int x;
34             scanf("%d",&x);
35             printf("%d",sum(tree, position[x]-1));
36             update(tree, position[x], -1);
37             position[x] = i;
38             update(tree, position[x], 1);
39         }
40     }
41     return 0;
42 }
```

*Handwritten notes:*

- Line 24: ~~N~~ N+M
- Line 26:  $M+1$   $M+N$
- Line 29:  $m+i$
- Line 30:  $m+i$
- Line 32:  $\sim$
- Line 34:  $x$
- Line 35:  $\text{sum}$   $\text{position}[x]-1$
- Line 36:  $-1$   $\rightarrow 0$
- Line 38:  $1$   $0 \rightarrow 1$
- Bottom:  $M, N (N+M)$

결과	메모리	시간	코드 길이
맞았습니다!!	2408 KB	128 ms	922 B



C++14

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 #include <map>
5 using namespace std;
6 void update(vector<int> &tree, int i, int diff) {
7     while (i < tree.size()) {
8         tree[i] += diff;
9         i += (i & -i);
10    }
11 }
12 int sum(vector<int> &tree, int i) {
13     int ans = 0;
14     while (i > 0) {
15         ans += tree[i];
16         i -= (i & -i);
17     }
18     return ans;
19 }
20 int sum(vector<int> &tree, int l, int r) {
21     if (l > r) return 0;
22     return sum(tree, r) - sum(tree, l-1);
23 }
24 int main() {
25     ios_base::sync_with_stdio(false);
26     cin.tie(nullptr);
27     int n;
28     cin >> n;
29     vector<int> a(n+1);
30     vector<int> tree(n+1);
31     for (int i=1; i<=n; i++) {
32         cin >> a[i];
33         if (a[i] % 2 == 0) {
34             update(tree, i, 1);
35         }
36     }
37     int m;
38     cin >> m;
39     while (m--) {
40         int w;
41         cin >> w;
42         if (w == 1) {
43             int index, val;
44             cin >> index >> val;
45             if (a[index]%2 == 0 && val%2 == 1) { //even -> odd
46                 update(tree, index, -1);
47             }
48             if (a[index]%2 == 1 && val%2 == 0) { //odd -> even
49                 update(tree, index, 1);
50             }
51             a[index] = val;
52         } else {
53             int l, r;
54             cin >> l >> r;
55             int cnt = sum(tree, l, r);
56             if (w == 3) {
57                 cnt = (r-l+1) - cnt;
58             }
59             cout << cnt << '\n';
60         }
61     }
62     return 0;
63 }
```

정답 !  
정답 0  
NlogN  
(N+M)logN

AC [index] = val;  
정답 -> 정답  
even -> odd  
1 -> 0  
odd -> even  
0 -> 1

24: [0,1]  
정답 정답  
34: [0,1]  
정답

C++14

```
1 #include <cstdio>
2 #include <cmath>
3 #include <cassert>
4 #include <vector>
5 #include <algorithm>
6 using namespace std;
7 void init(vector<int> &tree, vector<int> &a, int node, int start, int end) {
8     if (start == end) {
9         tree[node] = a[start];
10        return;
11    }
12    init(tree, a, node*2, start, (start+end)/2);
13    init(tree, a, node*2+1, (start+end)/2+1, end);
14    tree[node] = tree[node*2] * tree[node*2+1];
15 }
16 int sign(vector<int> &tree, int node, int start, int end, int left, int right) {
17     if (left > end || right < start) {
18         return 1;
19     }
20     if (left <= start && end <= right) {
21         return tree[node];
22     }
23     return sign(tree, node*2, start, (start+end)/2, left, right) * sign(tree, node*2+1, (start+end)/2+1, end,
left, right);
24 }
25 void update(vector<int> &tree, int node, int start, int end, int i, int value) {
26     if (i > end || i < start) {
27         return;
28     }
29     if (start == end) {
30         tree[node] = value;
31         return;
32     }
33     update(tree, node*2, start, (start+end)/2, i, value);
34     update(tree, node*2+1, (start+end)/2+1, end, i, value);
35     tree[node] = tree[node*2] * tree[node*2+1];
36 }
37 int main() {
38     int n,m;
39     while (scanf("%d %d",&n,&m) == 2) {
40         vector<int> a(n);
41         for (int i=0; i<n; i++) {
42             scanf("%d",&a[i]);
43             if (a[i] > 0) a[i] = 1;
44             if (a[i] < 0) a[i] = -1;
45         }
46         int h = (int)ceil(log2(n));
47         int tree_size = (1 << (h+1));
48         vector<int> tree(tree_size);
49         init(tree, a, 1, 0, n-1);
50         while (m--) {
51             char cmd;
52             scanf(" %c",&cmd);
53             if (cmd == 'C') {
54                 int index, value;
55                 scanf("%d %d",&index,&value);
56                 if (value > 0) value = 1;
57                 if (value < 0) value = -1;
58                 update(tree, 1, 0, n-1, index-1, value);
59             } else {
60                 int start, end;
61                 scanf("%d %d",&start,&end);
62                 int ans = sign(tree, 1, 0, n-1, start-1, end-1);
63                 if (ans > 0) {
64                     putchar('+');
65                 } else if (ans < 0) {
66                     putchar('-');
67                 } else {
68                     putchar('0');
69                 }
70             }
71         }
72         puts("");
73     }
74     return 0;
75 }
```

결과	메모리	시간	코드 길이
맞았습니다!!!	2668 KB	164 ms	2290 B

C++14

```
1 #include <cstdio>
2 #include <cmath>
3 #include <cassert>
4 #include <vector>
5 #include <algorithm>
6 using namespace std;
7 const int inf = 200000;
8 int sum(vector<int> &tree, int i) {
9     int ans = 0;
10    while (i > 0) {
11        ans += tree[i];
12        i -= i&-i;
13    }
14    return ans;
15 }
16 int sum(vector<int> &tree, int start, int end) {
17     return sum(tree, end) - sum(tree, start-1);
18 }
19 void update(vector<int> &tree, int i, int diff) {
20     while (i < tree.size()) {
21         tree[i] += diff;
22         i += i&-i;
23     }
24 }
25 int main() {
26     int n,m;
27     while (scanf("%d %d",&n,&m) == 2) {
28         vector<int> a(n+1);
29         vector<int> tree(n+1);
30         for (int i=1; i<=n; i++) {
31             scanf("%d",&a[i]);
32             if (a[i] < 0) {
33                 update(tree, i, 1);
34             } else if (a[i] == 0) {
35                 update(tree, i, inf);
36             }
37         }
38         while (m--) {
39             char cmd;
40             scanf(" %c",&cmd);
41             if (cmd == 'C') {
42                 int index, value;
43                 scanf("%d %d",&index,&value);
44                 if (a[index] < 0) {
45                     update(tree, index, -1);
46                 } else if (a[index] == 0) {
47                     update(tree, index, -inf);
48                 }
49                 a[index] = value;
50                 if (a[index] < 0) {
51                     update(tree, index, 1);
52                 } else if (a[index] == 0) {
53                     update(tree, index, inf);
54                 }
55             } else if (cmd == 'P') {
56                 int start, end;
57                 scanf("%d %d",&start, &end);
58                 int cnt = sum(tree, start, end);
59                 if (cnt > inf) {
60                     putchar('0');
61                 } else if (cnt % 2 == 0) {
62                     putchar('+');
63                 } else {
64                     putchar('-');
65                 }
66             }
67         }
68         puts("");
69     }
70     return 0;
71 }
```

$N \leq 10^5$

$inf = 2 \times 10^5$   
200000

결과

메모리

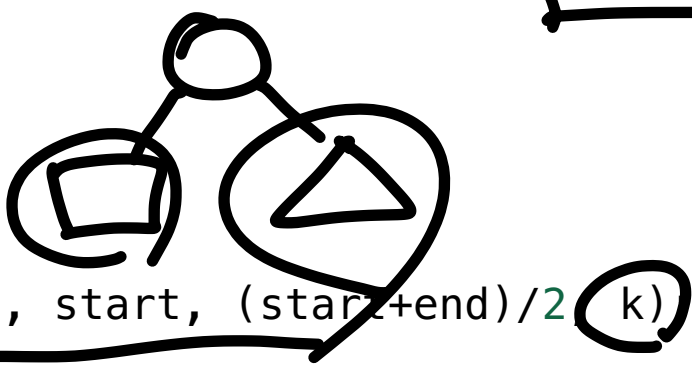
시간

코드 길이



C++14

```
1 #include <cstdio>
2 #include <vector>
3 #include <cmath>
4 using namespace std;
5 const int MAX = 1000000;
6 int init(vector<int> &a, vector<int> &tree, int node, int start, int end) {
7     if (start == end) {
8         return tree[node] = a[start];
9     } else {
10         return tree[node] = init(a, tree, node*2, start, (start+end)/2) + init(a, tree, node*2+1,
11                                     (start+end)/2+1, end);
12     }
13 }
14 void update(vector<int> &tree, int node, int start, int end, int i, int diff) {
15     if (i < start || i > end) return;
16     tree[node] = tree[node] + diff;
17     if (start != end) {
18         update(tree, node*2, start, (start+end)/2, i, diff);
19         update(tree, node*2+1, (start+end)/2+1, end, i, diff);
20     }
21 }
22 int kth(vector<int> &tree, int node, int start, int end, int k) {
23     if (start == end) {
24         return start;
25     } else {
26         if (k <= tree[node*2]) {
27             return kth(tree, node*2, start, (start+end)/2, k);
28         } else {
29             return kth(tree, node*2+1, (start+end)/2+1, end, k - tree[node*2]);
30         }
31     }
32 }
33 int main() {
34     vector<int> a(MAX+1);
35     int h = (int)ceil(log2(MAX+1));
36     int tree_size = (1 << (h+1));
37     vector<int> tree(tree_size);
38     int m;
39     scanf("%d", &m);
40     while (m--) {
41         int cmd;
42         scanf("%d", &cmd);
43         if (cmd == 2) {
44             int box, diff;
45             scanf("%d %d", &box, &diff);
46             a[box] += diff;
47             update(tree, 1, 1, MAX, box, diff);
48         } else if (cmd == 1) {
49             int k;
50             scanf("%d", &k);
51             int box = kth(tree, 1, 1, MAX, k);
52             printf("%d\n", box);
53             a[box] -= 1;
54             update(tree, 1, 1, MAX, box, -1);
55         }
56     }
57     return 0;
58 }
```



MAX = 10<sup>6</sup>

log MAX

C++14

```
1 #include <cstdio>
2 #include <cmath>
3 #include <cassert>
4 #include <vector>
5 #include <algorithm>
6 using namespace std;
7 void init(vector<int> &tree, int node, int start, int end) {
8     if (start == end) {
9         tree[node] = 1;
10        return;
11    }
12    init(tree, node*2, start, (start+end)/2);
13    init(tree, node*2+1, (start+end)/2+1, end);
14    tree[node] = tree[node*2] + tree[node*2+1];
15 }
16 void update(vector<int> &tree, int node, int start, int end, int i, int diff) {
17     if (i > end || i < start) {
18         return;
19     }
20     if (start == end) {
21         tree[node] += diff;
22         return;
23     }
24     update(tree, node*2, start, (start+end)/2, i, diff);
25     update(tree, node*2+1, (start+end)/2+1, end, i, diff);
26     tree[node] = tree[node*2] + tree[node*2+1];
27 }
28 int kth(vector<int> &tree, int node, int start, int end, int k) {
29     if (start == end) {
30         return start;
31     }
32     if (k <= tree[node*2]) {
33         return kth(tree, node*2, start, (start+end)/2, k);
34     } else {
35         return kth(tree, node*2+1, (start+end)/2+1, end, k-tree[node*2]);
36     }
37 }
38 int main() {
39     int n;
40     scanf("%d",&n);
41     int h = (int)ceil(log2(n));
42     int tree_size = (1 << (h+1));
43     vector<int> a(n+1);
44     vector<int> ans(n+1);
45     vector<int> tree(tree_size);
46     for (int i=1; i<=n; i++) {
47         scanf("%d",&a[i]);
48     }
49     init(tree, 1, 1, n);
50     for (int i=1; i<=n; i++) {
51         int k = a[i];
52         int w = kth(tree, 1, 1, n, k);
53         ans[w] = i;
54         update(tree, 1, 1, n, w, -1);
55     }
56     for (int i=1; i<=n; i++) {
57         printf("%d\n",ans[i]);
58     }
59     return 0;
60 }
```

변경  
0: 2492

$O(N \log N)$

끝

---

# 코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 [codeplus@startlink.io](mailto:codeplus@startlink.io) 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.