

시그 트리와 편향 트리 (연습)

소스 코드

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

---

C++14

```
1 #include <iostream>
2 #include <cmath>
3 #include <vector>
4 #include <algorithm>
5 using namespace std;
6 void init(vector<int> &a, vector<int> &tree, int node, int start, int end) {
7     if (start == end) {
8         tree[node] = start;
9     } else {
10         init(a, tree, node*2, start, (start+end)/2);
11         init(a, tree, node*2+1, (start+end)/2+1, end);
12         if (a[tree[node*2]] < a[tree[node*2+1]]) {
13             tree[node] = tree[node*2];
14         } else {
15             tree[node] = tree[node*2+1];
16         }
17     }
18 }
19 int query(vector<int> &a, vector<int> &tree, int node, int start, int end, int i, int j) {
20     if (i > end || j < start) {
21         return -1;
22     }
23     if (i <= start && end <= j) {
24         return tree[node];
25     }
26     int m1 = query(a, tree, 2*node, start, (start+end)/2, i, j);
27     int m2 = query(a, tree, 2*node+1, (start+end)/2+1, end, i, j);
28     if (m1 == -1) {
29         return m2;
30     } else if (m2 == -1) {
31         return m1;
32     } else {
33         if (a[m1] <= a[m2]) {
34             return m1;
35         } else {
36             return m2;
37         }
38     }
39 }
40 long long largest(vector<int> &a, vector<int> &tree, int start, int end) {
41     int n = a.size();
42     int m = query(a, tree, 1, 0, n-1, start, end);
43     long long area = (long long)(end-start+1)*(long long)a[m];
44     if (start <= m-1) {
45         long long temp = largest(a, tree, start, m-1);
46         if (area < temp) {
47             area = temp;
48         }
49     }
50     if (m+1 <= end) {
51         long long temp = largest(a, tree, m+1, end);
52         if (area < temp) {
53             area = temp;
54         }
55     }
56     return area;
57 }
58 int main() {
59     while (true) {
60         int n;
61         cin >> n;
62         if (n == 0) break;
63         vector<int> a(n);
64         for (int i=0; i<n; i++) {
65             cin >> a[i];
66         }
67         int h = (int)(ceil(log2(n))+1e-9);
68         int tree_size = (1 << (h+1));
69         vector<int> tree(tree_size);
70         init(a, tree, 1, 0, n-1);
71         cout << largest(a, tree, 0, n-1) << '\n';
72     }
73     return 0;
74 }
```

결과

메모리

시간

코드 길이

C++14

```
1 #include <iostream>
2 #include <cmath>
3 #include <vector>
4 #include <algorithm>
5 using namespace std;
6 struct SegmentTree {
7     vector<int> tree;
8     vector<int> a;
9     int n;
10    SegmentTree(vector<int> _a) {
11        a = _a;
12        n = a.size();
13        int h = (int)ceil(log2(n+1));
14        int tree_size = (1 << (h+1));
15        tree.resize(tree_size);
16        init(1, 0, n-1);
17    }
18    void init(int node, int start, int end) {
19        if (start == end) {
20            tree[node] = start;
21            return;
22        }
23        int mid = (start+end)/2;
24        init(node*2, start, mid);
25        init(node*2+1, mid+1, end);
26        if (a[tree[node*2]] > a[tree[node*2+1]]) {
27            tree[node] = tree[node*2];
28        } else {
29            tree[node] = tree[node*2+1];
30        }
31    }
32    void update(int index, int val) {
33        update(1, 0, n-1, index, val);
34    }
35    void update(int node, int start, int end, int index, int val) {
36        if (index < start || end < index) {
37            return;
38        }
39        if (start == end) {
40            a[index] = val;
41            tree[node] = start;
42            return;
43        }
44        int mid = (start+end)/2;
45        update(node*2, start, mid, index, val);
46        update(node*2+1, mid+1, end, index, val);
47        if (a[tree[node*2]] > a[tree[node*2+1]]) {
48            tree[node] = tree[node*2];
49        } else {
50            tree[node] = tree[node*2+1];
51        }
52    }
53    int query(int i, int j) {
54        return query(1, 0, n-1, i, j);
55    }
56    int query(int node, int start, int end, int i, int j) {
57        if (end < i || j < start) return -1;
58        if (i <= start && end <= j) {
59            return tree[node];
60        }
61        int mid = (start+end)/2;
62        int m1 = query(node*2, start, mid, i, j);
63        int m2 = query(node*2+1, mid+1, end, i, j);
64        if (m1 == -1) {
65            return m2;
66        } else if (m2 == -1) {
67            return m1;
68        } else {
69            if (a[m1] > a[m2]) {
70                return m1;
71            } else {
72                return m2;
73            }
74        }
75    }
76 };
77 int main() {
78     ios_base::sync_with_stdio(false);
79     cin.tie(nullptr);
80     int n;
81     cin >> n;
82     vector<int> a(n);
83     for (int i=0; i<n; i++) {
84         cin >> a[i];
85     }
86     SegmentTree tree(a);
87     int m;
88     cin >> m;
89     while (m--) {
90         int w;
91         cin >> w;
92         if (w == 1) {
93             int index, val;
94             cin >> index >> val;
95             index -= 1;
96             a[index] = val;
97             tree.update(index, val);
98         } else {
99             int l, r;
100            cin >> l >> r;
101            l -= 1; r -= 1;
102            int ans = 0;
103            int p = tree.query(l, r);
104            if (p-1 >= l) {
105                int p2 = tree.query(l, p-1);
106                ans = max(ans, a[p] + a[p2]);
107            }
108            if (p+1 <= r) {
109                int p2 = tree.query(p+1, r);
110                ans = max(ans, a[p] + a[p2]);
111            }
112            cout << ans << '\n';
113        }
114    }
115    return 0;
116 }
117
```

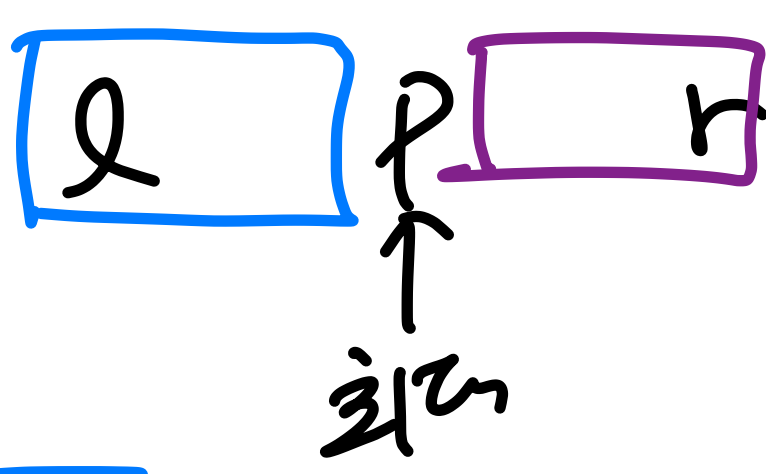
위치

tree[node]

Q

Σ'.  
O(MlogN)

logN



결과: 34

결과

메모리

시간

코드 길이

맞았습니다!!

4220 KB

120 ms

3021 B

C++14

```
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     vector<int> b(n);
30     for (int i=0; i<n; i++) {
31         scanf("%d",&a[i]);
32         b[i] = a[i];
33     }
34     sort(b.begin(), b.end());
35     b.erase(unique(b.begin(), b.end()), b.end());
36     map<int, int> d;
37     for (int i=0; i<b.size(); i++) {
38         d[b[i]] = i+1;
39     }
40     for (int i=0; i<n; i++) {
41         a[i] = d[a[i]];
42     }
43     long long ans = 0;
44     vector<int> tree(n+1);
45     l = n;
46     for (int i=0; i<n; i++) {
47         ans += (long long)sum(tree, a[i]+1, n);
48         update(tree, a[i], 1);
49     }
50     printf("%lld\n",ans);
51     return 0;
52 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

30464 KB

612 ms

1099 B



C++14

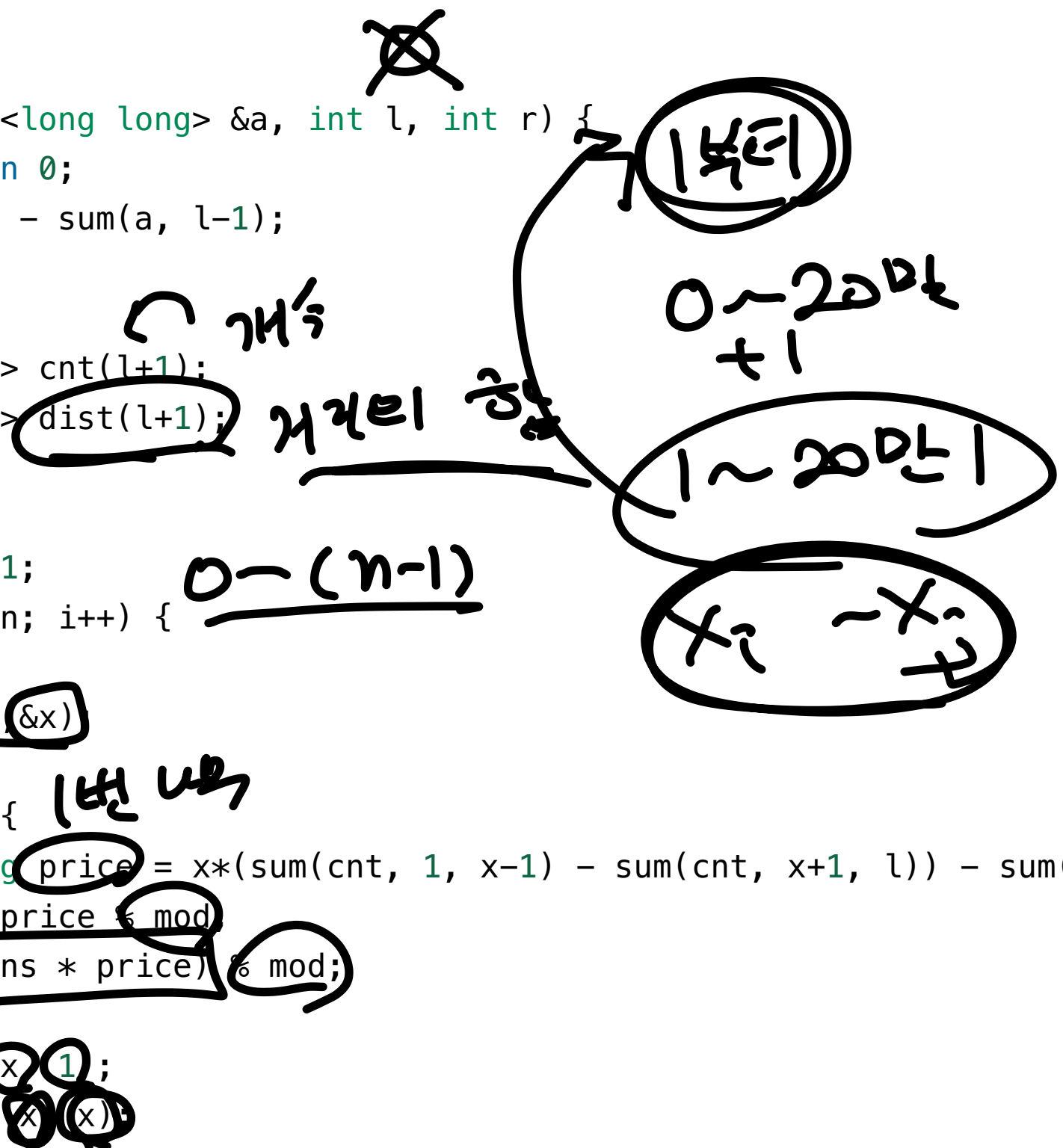
```
1 #include <cstdio>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5 int sum(vector<int> &tree, int i) {
6     int ans = 0;
7     while (i > 0) {
8         ans += tree[i];
9         i -= i&-i;
10    }
11    return ans;
12 }
13 int sum(vector<int> &tree, int start, int end) {
14     return sum(tree, end) - sum(tree, start-1);
15 }
16 void update(vector<int> &tree, int i, int diff) {
17     while (i < tree.size()) {
18         tree[i] += diff;
19         i += i&-i;
20    }
21 }
22 int main() {
23     int n;
24     scanf("%d",&n);
25     vector<pair<int,int>> a(n+1);
26     vector<int> tree(n+1);
27     for (int i=1; i<=n; i++) {
28         scanf("%d",&a[i].first);
29         a[i].second = 1;
30         update(tree, i, 1);
31     }
32     sort(a.begin()+1, a.end());
33     long long total = 0;
34     int last = 0;
35     vector<long long> ans(n+1);
36     int remain = n;
37     for (int i=1; i<=n; i++) {
38         total += (long long)(a[i].first - last)*remain;
39         remain -= 1;
40         last = a[i].first;
41         ans[a[i].second] = total - ((long long)sum(tree, a[i].second+1, n));
42         update(tree, a[i].second, -1);
43     }
44     for (int i=1; i<=n; i++) {
45         printf("%lld\n",ans[i]);
46     }
47     return 0;
48 }
```

정렬: 각 작업이  
실행  
= 작업이 끝나는  
순서대로  
점수를  
구해야  
하기 때문

$O(N \log N)$

C++14

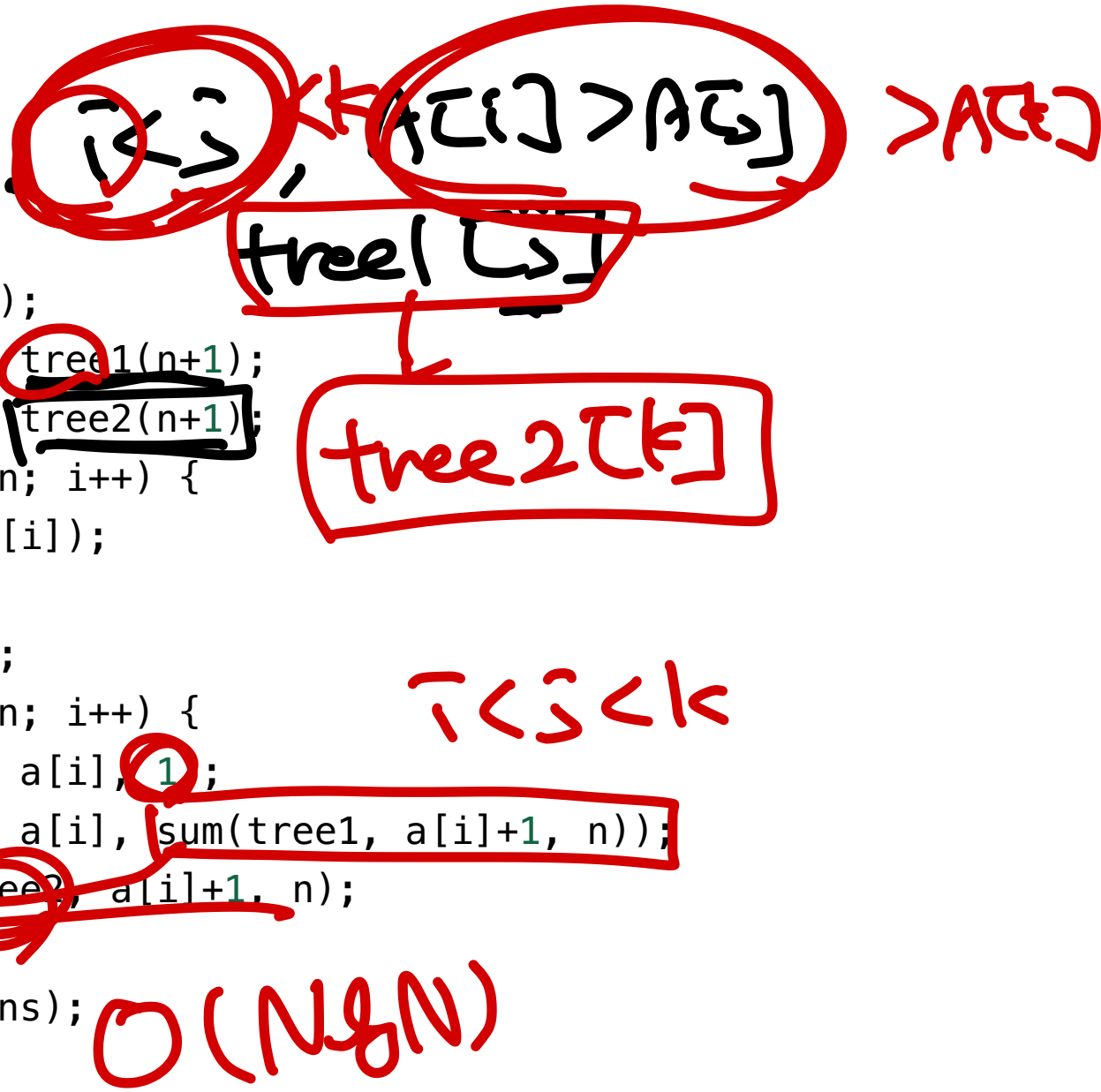
```
1 #include <cstdio>
2 #include <vector>
3 using namespace std;
4 long long mod = 1000000007LL;
5 int l = 200000;
6 void update(vector<long long> &a, int i, long long diff) {
7     while (i <= l) {
8         a[i] += diff;
9         i += (i & -i);
10    }
11 }
12 long long sum(vector<long long> &a, int i) {
13     long long ans = 0;
14     while (i > 0) {
15         ans += a[i];
16         i -= (i & -i);
17     }
18     return ans;
19 }
20 long long sum(vector<long long> &a, int l, int r) {
21     if (l > r) return 0;
22     return sum(a, r) - sum(a, l-1);
23 }
24 int main() {
25     vector<long long> cnt(l+1);
26     vector<long long> dist(l+1);
27     int n;
28     scanf("%d",&n);
29     long long ans = 1;
30     for (int i=0; i<n; i++) {
31         long long x;
32         scanf("%lld",&x);
33         x += 1;
34         if (x != 0) {
35             long long price = x*(sum(cnt, 1, x-1) - sum(cnt, x+1, l)) - sum(dist, 1, x-1) + sum(dist, x+1, l);
36             price = price % mod;
37             ans = (ans * price) % mod;
38         }
39         update(cnt, x, 1);
40         update(dist, x, x);
41     }
42     printf("%lld\n",ans);
43     return 0;
44 }
```



결과	메모리	시간	코드 길이
맞았습니다!!	4356 KB	72 ms	1048 B

C++14

```
1 #include <cstdio>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5 long long sum(vector<long long> &tree, int i) {
6     long long ans = 0;
7     while (i > 0) {
8         ans += tree[i];
9         i -= i&-i;
10    }
11    return ans;
12 }
13 long long sum(vector<long long> &tree, int start, int end) {
14     if (start > end) return 0;
15     return sum(tree, end) - sum(tree, start-1);
16 }
17 void update(vector<long long> &tree, int i, long long diff) {
18     while (i < tree.size()) {
19         tree[i] += diff;
20         i += i&-i;
21     }
22 }
23 int main() {
24     int n;
25     scanf("%d",&n);
26     vector<int> a(n+1);
27     vector<long long> tree1(n+1);
28     vector<long long> tree2(n+1);
29     for (int i=1; i<=n; i++) {
30         scanf("%d",&a[i]);
31     }
32     long long ans = 0;
33     for (int i=1; i<=n; i++) {
34         update(tree1, a[i], 1);
35         update(tree2, a[i], sum(tree1, a[i]+1, n));
36         ans += sum(tree2, a[i]+1, n);
37     }
38     printf("%lld\n",ans);
39     return 0;
40 }
```



결과	메모리	시간	코드 길이
맞았습니다!!	3188 KB	24 ms	945 B

C++14

```
1 #include <iostream>
2 #include <cmath>
3 #include <vector>
4 #include <algorithm>
5 using namespace std;
6 void init(vector<int> &tree, int node, int start, int end) {
7     if (start == end) {
8         tree[node] = 1;
9         return;
10    }
11    init(tree, node*2, start, (start+end)/2);
12    init(tree, node*2+1, (start+end)/2+1, end);
13    tree[node] = tree[node*2] + tree[node*2+1];
14 }
15 int kth(vector<int> &tree, int node, int start, int end, int k) {
16     if (start == end) {
17         tree[node] = 0;
18         return start;
19     }
20     int ans = 0;
21     if (k <= tree[node*2]) {
22         ans = kth(tree, node*2, start, (start+end)/2, k);
23     } else {
24         ans = kth(tree, node*2+1, (start+end)/2+1, end, k-tree[node*2]);
25     }
26     tree[node] = tree[node*2] + tree[node*2+1];
27     return ans;
28 }
29 int main() {
30     ios_base::sync_with_stdio(false);
31     cin.tie(nullptr);
32     int n;
33     cin >> n;
34     vector<int> b(n), g(n), w(n);
35     for (int i=0; i<n; i++) {
36         cin >> b[i];
37     }
38     for (int i=0; i<n; i++) {
39         cin >> g[i];
40     }
41     for (int i=0; i<n; i++) {
42         cin >> w[i];
43     }
44     int height = 0;
45     for (int i=0; i<n; i++) {
46         if (height < b[i]+g[i]+w[i]+1) {
47             height = b[i]+g[i]+w[i]+1;
48         }
49     }
50     vector<int> y(height+1);
51     y[0] = 0;
52     y[1] = n;
53     for (int i=0; i<n; i++) {
54         y[b[i]+1] += 1;
55         y[b[i]+g[i]+1] += 3;
56         y[b[i]+g[i]+w[i]+1] -= 5;
57     }
58     vector<int> x(height+1);
59     for (int i=1; i<=height; i++) {
60         x[i] = x[i-1] + y[i];
61     }
62     int h = (int)ceil(log2(height));
63     int tree_size = (1 << (h+1));
64     vector<int> tree(tree_size);
65     init(tree, 1, 1, height);
66     int m;
67     cin >> m;
68     while (m--) {
69         int h;
70         cin >> h;
71         int k = kth(tree, 1, 1, height, h);
72         cout << x[k] << '\n';
73     }
74     return 0;
75 }
76
```

결과	메모리	시간	코드 길이
맞았습니다!!!	59804 KB	288 ms	1812 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 MAX = 65536;
8 void update(vector<int> &tree, int node, int start, int end, int i, int diff) {
9     if (i > end || i < start) {
10         return;
11     }
12     if (start == end) {
13         tree[node] += diff;
14         return;
15     }
16     update(tree, node*2, start, (start+end)/2, i, diff);
17     update(tree, node*2+1, (start+end)/2+1, end, i, diff);
18     tree[node] = tree[node*2] + tree[node*2+1];
19 }
20 int kth(vector<int> &tree, int node, int start, int end, int k) {
21     if (start == end) {
22         return start;
23     }
24     if (k <= tree[node*2]) {
25         return kth(tree, node*2, start, (start+end)/2, k);
26     } else {
27         return kth(tree, node*2+1, (start+end)/2+1, end, k-tree[node*2]);
28     }
29 }
30 int main() {
31     int n,k;
32     scanf("%d %d",&n,&k);
33     int h = (int)ceil(log2(MAX));
34     int tree_size = (1 << (h+1));
35     vector<int> a(n);
36     vector<int> tree(tree_size);
37     for (int i=0; i<n; i++) {
38         scanf("%d",&a[i]);
39     }
40     for (int i=0; i<k-1; i++) {
41         update(tree, 1, 0, MAX-1, a[i], 1);
42     }
43     long long ans = 0;
44     for (int i=k-1; i<n; i++) {
45         update(tree, 1, 0, MAX-1, a[i], 1);
46         ans += (long long)kth(tree, 1, 0, MAX-1, (k+1)/2);
47         update(tree, 1, 0, MAX-1, a[i-k+1], -1);
48     }
49     printf("%lld\n",ans);
50     return 0;
51 }
```

$N \log M$   
 $O(N \log M)$

결과

메모리

시간

코드 길이

맞았습니다!!

2724 KB

132 ms

1386 B

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 sum(vector<int> &tree, int node, int start, int end, int i, int j) {
29     if (i > end || j < start) {
30         return 0;
31     }
32     if (i <= start && end <= j) {
33         return tree[node];
34     }
35     return sum(tree, node*2, start, (start+end)/2, i, j) + sum(tree, node*2+1, (start+end)/2+1, end, i, j);
36 }
37 int kth(vector<int> &tree, int node, int start, int end, int k) {
38     if (start == end) {
39         return start;
40     }
41     if (k <= tree[node*2]) {
42         return kth(tree, node*2, start, (start+end)/2, k);
43     } else {
44         return kth(tree, node*2+1, (start+end)/2+1, end, k-tree[node*2]);
45     }
46 }
47 int main() {
48     int n,k;
49     scanf("%d %d",&n,&k);
50     int h = (int)ceil(log2(n));
51     int tree_size = (1 << (h+1));
52     vector<int> ans(n+1);
53     vector<int> tree(tree_size);
54     init(tree, 1, 1, n);
55     int last = 0;
56     printf("<");
57     for (int i=n; i>=1; i--) {
58         int pre = 0;
59         if (last != 0) {
60             pre = sum(tree, 1, 1, n, 1, last);
61         }
62         int index = (pre - k % i);
63         if (index == 0) {
64             index = i;
65         }
66         last = kth(tree, 1, 1, n, index);
67         printf("%d", last);
68         if (i != 1) {
69             printf(", ");
70         }
71         update(tree, 1, 1, n, last, -1);
72     }
73     printf(">\n");
74     return 0;
75 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

2660 KB

60 ms

2044 B

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.