세그트리화 펜윅트리 (인덱스)

소스 코드

최백준 choi@startlink.io

C++14

```cpp
#include <iostream>
#include <cmath>
#include <vector>
#include <algorithm>
using namespace std;
void init(vector<int> &a, vector<int> &tree, int node, int start, int end) {
    if (start == end) {
        tree[node] = start;
    } else {
        init(a, tree, node*2, start, (start+end)/2);
        init(a, tree, node*2+1, (start+end)/2+1, end);
        if (a[tree[node*2]] <= a[tree[node*2+1]]) {
            tree[node] = tree[node*2];
        } else {
            tree[node] = tree[node*2+1];
        }
    }
}
int query(vector<int> &a, vector<int> &tree, int node, int start, int end, int i, int j) {
    if (i > end || j < start) {
        return -1;
    }
    if (i <= start && end <= j) {
        return tree[node];
    }
    int m1 = query(a, tree, 2*node, start, (start+end)/2, i, j);
    int m2 = query(a, tree, 2*node+1, (start+end)/2+1, end, i, j);
    if (m1 == -1) {
        return m2;
    } else if (m2 == -1) {
        return m1;
    } else {
        if (a[m1] <= a[m2]) {
            return m1;
        } else {
            return m2;
        }
    }
}
long long largest(vector<int> &a, vector<int> &tree, int start, int end) {
    int n = a.size();
    int m = query(a, tree, 1, 0, n-1, start, end);
    long long area = (long long)(end-start+1)*(long long)a[m];
    if (start <= m-1) {
        long long temp = largest(a, tree, start, m-1);
        if (area < temp) {
            area = temp;
        }
    }
    if (m+1 <= end) {
        long long temp = largest(a, tree, m+1, end);
        if (area < temp) {
            area = temp;
        }
    }
    return area;
}
int main() {
    while (true) {
        int n;
        cin >> n;
        if (n == 0) break;
        vector<int> a(n);
        for (int i=0; i<n; i++) {
            cin >> a[i];
        }
        int h = (int)(ceil(log2(n))+1e-9);
        int tree_size = (1 << (h+1));
        vector<int> tree(tree_size);
        init(a, tree, 1, 0, n-1);
        cout << largest(a, tree, 0, n-1) << '\n';
    }
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	9544 KB	216 ms	2000 B

C++14

```cpp
#include <iostream>
#include <cmath>
#include <vector>
#include <algorithm>
using namespace std;
struct SegmentTree {
    vector<int> tree;
    vector<int> a;
    int n;
    SegmentTree(vector<int> _a) {
        a = _a;
        n = a.size();
        int h = (int)ceil(log2(n+1));
        int tree_size = (1 << (h+1));
        tree.resize(tree_size);
        init(1, 0, n-1);
    }
    void init(int node, int start, int end) {
        if (start == end) {
            tree[node] = start;
            return;
        }
        int mid = (start+end)/2;
        init(node*2, start, mid);
        init(node*2+1, mid+1, end);
        if (a[tree[node*2]] > a[tree[node*2+1]]) {
            tree[node] = tree[node*2];
        } else {
            tree[node] = tree[node*2+1];
        }
    }
    void update(int index, int val) {
        update(1, 0, n-1, index, val);
    }
    void update(int node, int start, int end, int index, int val) {
        if (index < start || end < index) {
            return;
        }
        if (start == end) {
            a[index] = val;
            tree[node] = start;
            return;
        }
        int mid = (start+end)/2;
        update(node*2, start, mid, index, val);
        update(node*2+1, mid+1, end, index, val);
        if (a[tree[node*2]] > a[tree[node*2+1]]) {
            tree[node] = tree[node*2];
        } else {
            tree[node] = tree[node*2+1];
        }
    }
    int query(int i, int j) {
        return query(1, 0, n-1, i, j);
    }
    int query(int node, int start, int end, int i, int j) {
        if (end < i || j < start) return -1;
        if (i <= start && end <= j) {
            return tree[node];
        }
        int mid = (start+end)/2;
        int m1 = query(node*2, start, mid, i, j);
        int m2 = query(node*2+1, mid+1, end, i, j);
        if (m1 == -1) {
            return m2;
        } else if (m2 == -1) {
            return m1;
        } else {
            if (a[m1] > a[m2]) {
                return m1;
            } else {
                return m2;
            }
        }
    }
};
int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int n;
    cin >> n;
    vector<int> a(n);
    for (int i=0; i<n; i++) {
        cin >> a[i];
    }
    SegmentTree tree(a);
    int m;
    cin >> m;
    while (m--) {
        int w;
        cin >> w;
        if (w == 1) {
            int index, val;
            cin >> index >> val;
            index -= 1;
            a[index] = val;
            tree.update(index, val);
        } else {
            int l, r;
            cin >> l >> r;
            l -= 1; r -= 1;
            int ans = 0;
            int p = tree.query(l, r);
            if (p-1 >= l) {
                int p2 = tree.query(l, p-1);
                ans = max(ans, a[p] + a[p2]);
            }
            if (p+1 <= r) {
                int p2 = tree.query(p+1, r);
                ans = max(ans, a[p] + a[p2]);
            }
            cout << ans << '\n';
        }
    }
    return 0;
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	4220 KB	120 ms	3021 B

C++14

```cpp
#include <cstdio>
#include <vector>
#include <algorithm>
#include <map>
using namespace std;
int l;
void update(vector<int> &tree, int i, int diff) {
    while (i <= l) {
        tree[i] += diff;
        i += (i & -i);
    }
}
int sum(vector<int> &tree, int i) {
    int ans = 0;
    while (i > 0) {
        ans += tree[i];
        i -= (i & -i);
    }
    return ans;
}
int sum(vector<int> &tree, int l, int r) {
    if (l > r) return 0;
    return sum(tree, r) - sum(tree, l-1);
}
int main() {
    int n;
    scanf("%d",&n);
    vector<int> a(n);
    vector<int> b(n);
    for (int i=0; i<n; i++) {
        scanf("%d",&a[i]);
        b[i] = a[i];
    }
    sort(b.begin(), b.end());
    b.erase(unique(b.begin(), b.end()), b.end());
    map<int, int> d;
    for (int i=0; i<b.size(); i++) {
        d[b[i]] = i+1;
    }
    for (int i=0; i<n; i++) {
        a[i] = d[a[i]];
    }
    long long ans = 0;
    vector<int> tree(n+1);
    l = n;
    for (int i=0; i<n; i++) {
        ans += (long long)sum(tree, a[i]+1, n);
        update(tree, a[i], 1);
    }
    printf("%lld\n",ans);
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	30464 KB	612 ms	1099 B

C++14

```cpp
#include <cstdio>
#include <vector>
#include <algorithm>
using namespace std;
int sum(vector<int> &tree, int i) {
    int ans = 0;
    while (i > 0) {
        ans += tree[i];
        i -= i&-i;
    }
    return ans;
}
int sum(vector<int> &tree, int start, int end) {
    return sum(tree, end) - sum(tree, start-1);
}
void update(vector<int> &tree, int i, int diff) {
    while (i < tree.size()) {
        tree[i] += diff;
        i += i&-i;
    }
}
int main() {
    int n;
    scanf("%d",&n);
    vector<pair<int,int>> a(n+1);
    vector<int> tree(n+1);
    for (int i=1; i<=n; i++) {
        scanf("%d",&a[i].first);
        a[i].second = i;
        update(tree, i, 1);
    }
    sort(a.begin()+1, a.end());
    long long total = 0;
    int last = 0;
    vector<long long> ans(n+1);
    int remain = n;
    for (int i=1; i<=n; i++) {
        total += (long long)(a[i].first - last)*remain;
        remain -= 1;
        last = a[i].first;
        ans[a[i].second] = total - (long long)sum(tree, a[i].second+1, n);
        update(tree, a[i].second, -1);
    }
    for (int i=1; i<=n; i++) {
        printf("%lld\n",ans[i]);
    }
    return 0;
}
```

결과	메모리	시간	코드 길이
맞았습니다!!	3180 KB	56 ms	1144 B

C++14

```cpp
#include <cstdio>
#include <vector>
using namespace std;
long long mod = 1000000007LL;
int l = 200000;
void update(vector<long long> &a, int i, long long diff) {
    while (i <= l) {
        a[i] += diff;
        i += (i & -i);
    }
}
long long sum(vector<long long> &a, int i) {
    long long ans = 0;
    while (i > 0) {
        ans += a[i];
        i -= (i & -i);
    }
    return ans;
}
long long sum(vector<long long> &a, int l, int r) {
    if (l > r) return 0;
    return sum(a, r) - sum(a, l-1);
}
int main() {
    vector<long long> cnt(l+1);
    vector<long long> dist(l+1);
    int n;
    scanf("%d",&n);
    long long ans = 1;
    for (int i=0; i<n; i++) {
        long long x;
        scanf("%lld",&x);
        x += 1;
        if (i != 0) {
            long long price = x*(sum(cnt, 1, x-1) - sum(cnt, x+1, l)) - sum(dist,1, x-1) + sum(dist, x+1, l);
            price = price % mod;
            ans = (ans * price) % mod;
        }
        update(cnt, x, 1);
        update(dist, x, x);
    }
    printf("%lld\n",ans);
    return 0;
}
```

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

C++14

```cpp
#include <cstdio>
#include <vector>
#include <algorithm>
using namespace std;
long long sum(vector<long long> &tree, int i) {
    long long ans = 0;
    while (i > 0) {
        ans += tree[i];
        i -= i&-i;
    }
    return ans;
}
long long sum(vector<long long> &tree, int start, int end) {
    if (start > end) return 0;
    return sum(tree, end) - sum(tree, start-1);
}
void update(vector<long long> &tree, int i, long long diff) {
    while (i < tree.size()) {
        tree[i] += diff;
        i += i&-i;
    }
}
int main() {
    int n;
    scanf("%d",&n);
    vector<int> a(n+1);
    vector<long long> tree1(n+1);
    vector<long long> tree2(n+1);
    for (int i=1; i<=n; i++) {
        scanf("%d",&a[i]);
    }
    long long ans = 0;
    for (int i=1; i<=n; i++) {
        update(tree1, a[i], 1);
        update(tree2, a[i], sum(tree1, a[i]+1, n));
        ans += sum(tree2, a[i]+1, n);
    }
    printf("%lld\n",ans);
    return 0;
}
```

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

C++14

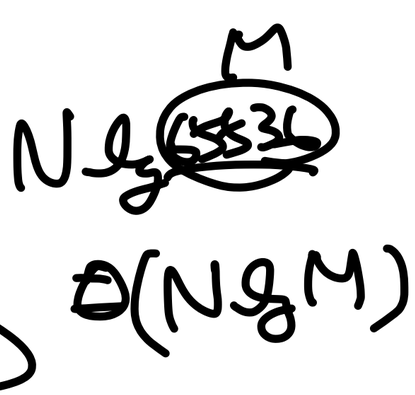
```cpp
#include <iostream>
#include <cmath>
#include <vector>
#include <algorithm>
using namespace std;
void init(vector<int> &tree, int node, int start, int end) {
    if (start == end) {
        tree[node] = 1;
        return;
    }
    init(tree, node*2, start, (start+end)/2);
    init(tree, node*2+1, (start+end)/2+1, end);
    tree[node] = tree[node*2] + tree[node*2+1];
}
int kth(vector<int> &tree, int node, int start, int end, int k) {
    if (start == end) {
        tree[node] = 0;
        return start;
    }
    int ans = 0;
    if (k <= tree[node*2]) {
        ans = kth(tree, node*2, start, (start+end)/2, k);
    } else {
        ans = kth(tree, node*2+1, (start+end)/2+1, end, k-tree[node*2]);
    }
    tree[node] = tree[node*2] + tree[node*2+1];
    return ans;
}
int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int n;
    cin >> n;
    vector<int> b(n), g(n), w(n);
    for (int i=0; i<n; i++) {
        cin >> b[i];
    }
    for (int i=0; i<n; i++) {
        cin >> g[i];
    }
    for (int i=0; i<n; i++) {
        cin >> w[i];
    }
    int height = 0;
    for (int i=0; i<n; i++) {
        if (height < b[i]+g[i]+w[i]+1) {
            height = b[i]+g[i]+w[i]+1;
        }
    }
    vector<int> y(height+1);
    y[0] = 0;
    y[1] = n;
    for (int i=0; i<n; i++) {
        y[b[i]+1] += 1;
        y[b[i]+g[i]+1] += 3;
        y[b[i]+g[i]+w[i]+1] -= 5;
    }
    vector<int> x(height+1);
    for (int i=1; i<=height; i++) {
        x[i] = x[i-1] + y[i];
    }
    int h = (int)ceil(log2(height));
    int tree_size = (1 << (h+1));
    vector<int> tree(tree_size);
    init(tree, 1, 1, height);
    int m;
    cin >> m;
    while (m--) {
        int h;
        cin >> h;
        int k = kth(tree, 1, 1, height, h);
        cout << x[k] << '\n';
    }
    return 0;
}

```

결과	메모리	시간	코드 길이
맞았습니다!!	59804 KB	288 ms	1812 B

C++14

```cpp
#include <cstdio>
#include <cmath>
#include <cassert>
#include <vector>
#include <algorithm>
using namespace std;
const int MAX = 65536;
void update(vector<int> &tree, int node, int start, int end, int i, int diff) {
    if (i > end || i < start) {
        return;
    }
    if (start == end) {
        tree[node] += diff;
        return;
    }
    update(tree, node*2, start, (start+end)/2, i, diff);
    update(tree, node*2+1, (start+end)/2+1, end, i, diff);
    tree[node] = tree[node*2] + tree[node*2+1];
}
int kth(vector<int> &tree, int node, int start, int end, int k) {
    if (start == end) {
        return start;
    }
    if (k <= tree[node*2]) {
        return kth(tree, node*2, start, (start+end)/2, k);
    } else {
        return kth(tree, node*2+1, (start+end)/2+1, end, k-tree[node*2]);
    }
}
int main() {
    int n,k;
    scanf("%d %d",&n,&k);
    int h = (int)ceil(log2(MAX));
    int tree_size = (1 << (h+1));
    vector<int> a(n);
    vector<int> tree(tree_size);
    for (int i=0; i<n; i++) {
        scanf("%d",&a[i]);
    }
    for (int i=0; i<k-1; i++) {
        update(tree, 1, 0, MAX-1, a[i], 1);
    }
    long long ans = 0;
    for (int i=k-1; i<n; i++) {
        update(tree, 1, 0, MAX-1, a[i], 1);
        ans += (long long)kth(tree, 1, 0, MAX-1, (k+1)/2);
        update(tree, 1, 0, MAX-1, a[i-k+1], -1);
    }
    printf("%lld\n",ans);
    return 0;
}
```



결과	메모리	시간	코드 길이
맞았습니다!!	2724 KB	132 ms	1386 B

C++14

```cpp
#include <cstdio>
#include <cmath>
#include <cassert>
#include <vector>
#include <algorithm>
using namespace std;
void init(vector<int> &tree, int node, int start, int end) {
    if (start == end) {
        tree[node] = 1;
        return;
    }
    init(tree, node*2, start, (start+end)/2);
    init(tree, node*2+1, (start+end)/2+1, end);
    tree[node] = tree[node*2] + tree[node*2+1];
}
void update(vector<int> &tree, int node, int start, int end, int i, int diff) {
    if (i > end || i < start) {
        return;
    }
    if (start == end) {
        tree[node] += diff;
        return;
    }
    update(tree, node*2, start, (start+end)/2, i, diff);
    update(tree, node*2+1, (start+end)/2+1, end, i, diff);
    tree[node] = tree[node*2] + tree[node*2+1];
}
int sum(vector<int> &tree, int node, int start, int end, int i, int j) {
    if (i > end || j < start) {
        return 0;
    }
    if (i <= start && end <= j) {
        return tree[node];
    }
    return sum(tree, node*2, start, (start+end)/2, i, j) + sum(tree, node*2+1, (start+end)/2+1, end, i, j);
}
int kth(vector<int> &tree, int node, int start, int end, int k) {
    if (start == end) {
        return start;
    }
    if (k <= tree[node*2]) {
        return kth(tree, node*2, start, (start+end)/2, k);
    } else {
        return kth(tree, node*2+1, (start+end)/2+1, end, k-tree[node*2]);
    }
}
int main() {
    int n,k;
    scanf("%d %d",&n,&k);
    int h = (int)ceil(log2(n));
    int tree_size = (1 << (h+1));
    vector<int> ans(n+1);
    vector<int> tree(tree_size);
    init(tree, 1, 1, n);
    int last = 0;
    printf("<");
    for (int i=n; i>=1; i--) {
        int pre = 0;
        if (last != 0) {
            pre = sum(tree, 1, 1, n, 1, last);
        }
        int index = (pre + k) % i;
        if (index == 0) {
            index = i;
        }
        last = kth(tree, 1, 1, n, index);
        printf("%d", last);
        if (i != 1) {
            printf(", ");
        }
        update(tree, 1, 1, n, last, -1);
    }
    printf(">\n");
    return 0;
}
```

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 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.