

시그먼트 트리 (참고)

스크립트

최백준 choi@startlink.io

C++14

```
1 #include <iostream>
2 #include <cmath>
3 #include <algorithm>
4 #include <vector>
5 #include <queue>
6 using namespace std;
7 vector<int> dfs_order;
8 vector<int> first;
9 vector<int> level;
10 vector<int> a[100001];
11 void init(vector<int> &tree, int node, int start, int end) {
12     if (start == end) {
13         tree[node] = start;
14     } else {
15         init(tree, node*2, start, (start+end)/2);
16         init(tree, node*2+1, (start+end)/2+1, end);
17         if (level[tree[node*2]] < level[tree[node*2+1]]) {
18             tree[node] = tree[node*2];
19         } else {
20             tree[node] = tree[node*2+1];
21         }
22     }
23 }
24 int query(vector<int> &tree, int node, int start, int end, int i, int j) {
25     if (i > end || j < start) {
26         return -1;
27     }
28     if (i <= start && end <= j) {
29         return tree[node];
30     }
31     int m1 = query(tree, 2*node, start, (start+end)/2, i, j);
32     int m2 = query(tree, 2*node+1, (start+end)/2+1, end, i, j);
33     if (m1 == -1) {
34         return m2;
35     } else if (m2 == -1) {
36         return m1;
37     } else {
38         if (level[m1] < level[m2]) {
39             return m1;
40         } else {
41             return m2;
42         }
43     }
44 }
45 void dfs(int node, int parent, int depth) {
46     dfs_order.push_back(node);
47     level.push_back(depth);
48     for (int child : a[node]) {
49         if (child == parent) continue;
50         dfs(child, node, depth+1);
51         dfs_order.push_back(node);
52         level.push_back(depth);
53     }
54 }
55 int main() {
56     ios_base::sync_with_stdio(false);
57     cin.tie(nullptr);
58     int n;
59     cin >> n;
60     for (int i=0; i<n-1; i++) {
61         int u, v;
62         cin >> u >> v;
63         a[u].push_back(v);
64         a[v].push_back(u);
65     }
66     first.resize(n+1, -1);
67     dfs(1, 0, 0);
68     for (int i=0; i<dfs_order.size(); i++) {
69         int node = dfs_order[i];
70         if (first[node] == -1) {
71             first[node] = i;
72         }
73     }
74     n = dfs_order.size();
75     int h = (int)ceil(log2(n));
76     int tree_size = (1 << (h+1));
77     vector<int> tree(tree_size);
78     init(tree, 1, 0, n-1);
79     int m;
80     cin >> m;
81     while (m--) {
82         int u, v;
83         cin >> u >> v;
84         int i = first[u];
85         int j = first[v];
86         if (i > j) swap(i, j);
87         int lca = query(tree, 1, 0, n-1, i, j);
88         cout << dfs_order[lca] << '\n';
89     }
90     return 0;
91 }
92
```

tree[] = 최소값
의인정

m1 < m2

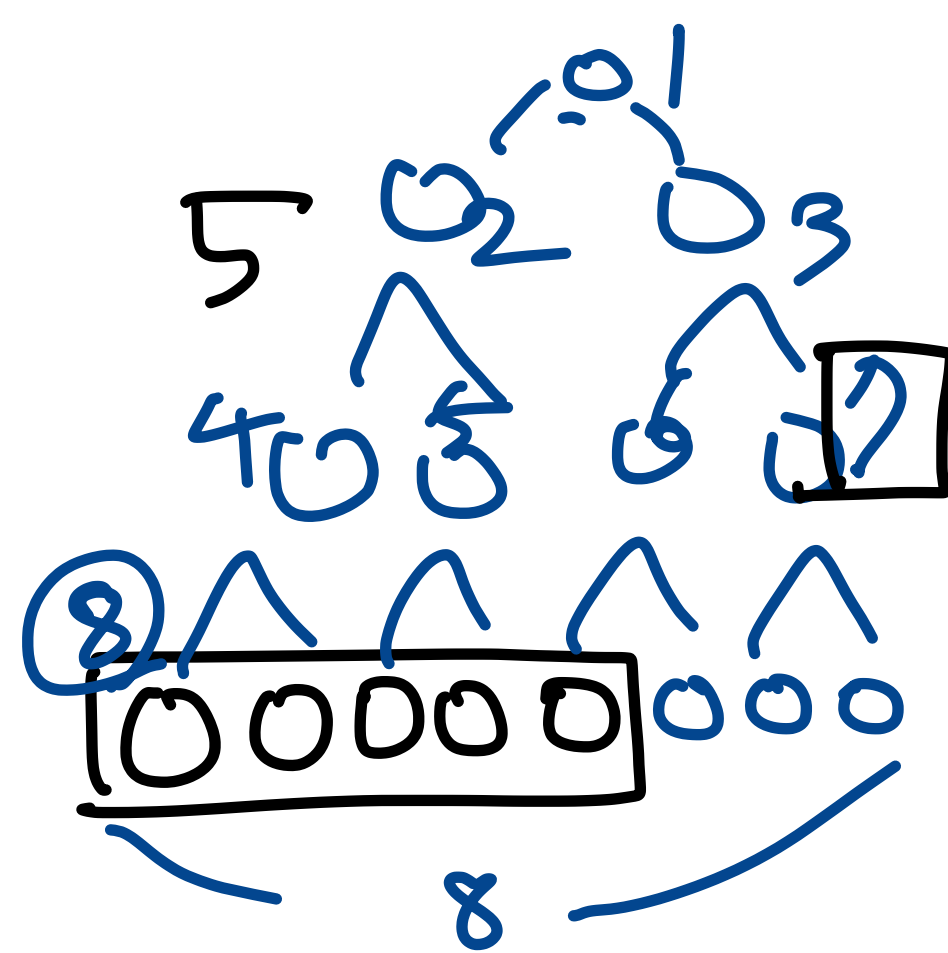
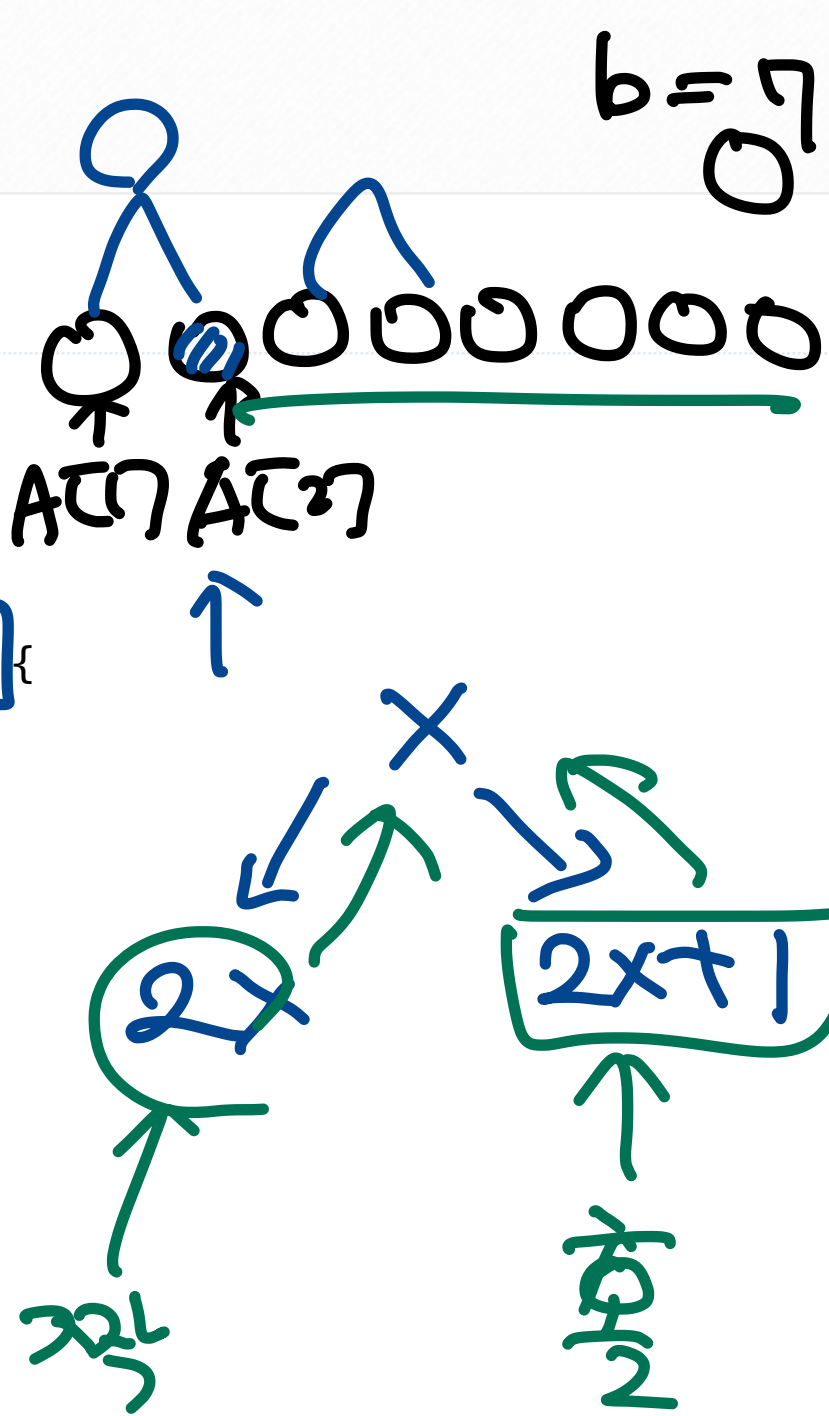
Parent → node
가장

first[]
= [번정정]을
2 방향한 편정

ST Query

C++14

```
1 #include <stdio>
2 long long tree[3000000];
3 int b = 1;
4 void update(int x, long long diff) {
5     x += b;
6     tree[x] += diff;
7     while (x > 1) {
8         tree[x/2] += diff;
9         x /= 2;
10    }
11 }
12 long long sum(int l, int r) {
13     l += b;
14     r += b;
15     long long ans = 0;
16     while (l < r) {
17         if (l%2 == 0) {
18             l /= 2;
19         } else {
20             ans += tree[l];
21             l += 1;
22             l /= 2;
23         }
24         if (r%2 == 1) {
25             r /= 2;
26         } else {
27             ans += tree[r];
28             r -= 1;
29             r /= 2;
30         }
31     }
32     if (l == r) {
33         ans += tree[l];
34     }
35     return ans;
36 }
37 int main() {
38     int n, m, k;
39     scanf("%d %d %d", &n, &m, &k);
40     m += k;
41     while (b <= n) {
42         b *= 2;
43     }
44     b -= 1;
45     for (int i=1; i<=n; i++) {
46         long long x;
47         scanf("%lld", &x);
48         update(i, x);
49     }
50     while (m-->0) {
51         int t1;
52         scanf("%d", &t1);
53         if (t1 == 1) {
54             int t2;
55             long long t3;
56             scanf("%d %lld", &t2, &t3);
57             update(t2, t3-tree[b+t2]);
58         } else if (t1 == 2) {
59             int t2, t3;
60             scanf("%d %d", &t2, &t3);
61             printf("%lld\n", sum(t2, t3));
62         }
63     }
64     return 0;
65 }
```



결과

메모리

시간

코드 길이

맞았습니다!!

24552 KB

228 ms

1244 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.