

트리 |

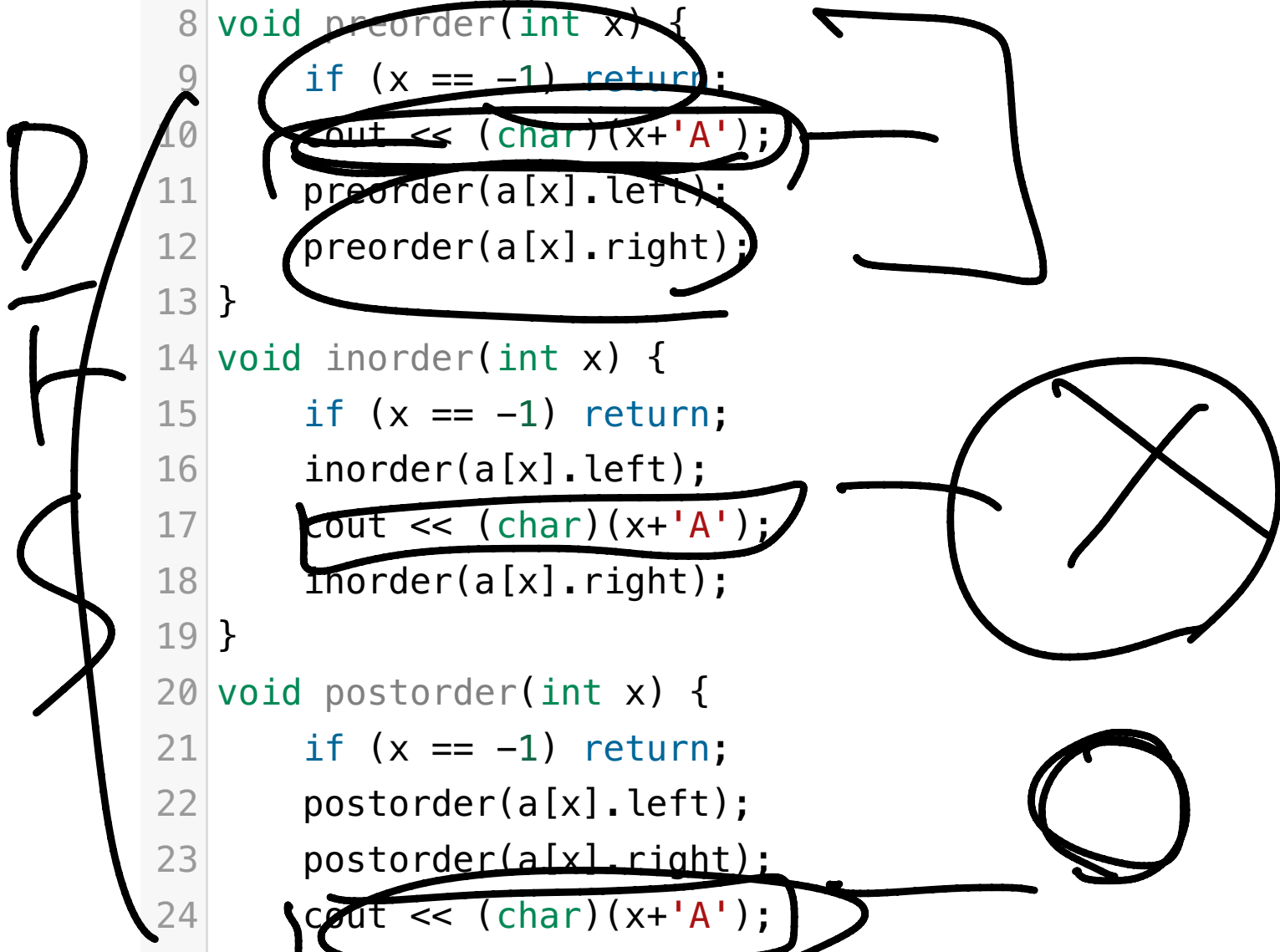
소스코드

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



C++14

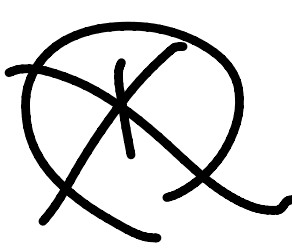
```
1 #include <iostream>
2 using namespace std;
3 struct Node {
4     int left;
5     int right;
6 };
7 Node a[50];
8 void preorder(int x) {
9     if (x == -1) return;
10    cout << (char)(x+'A');
11    preorder(a[x].left);
12    preorder(a[x].right);
13 }
14 void inorder(int x) {
15     if (x == -1) return;
16     inorder(a[x].left);
17     cout << (char)(x+'A');
18     inorder(a[x].right);
19 }
20 void postorder(int x) {
21     if (x == -1) return;
22     postorder(a[x].left);
23     postorder(a[x].right);
24     cout << (char)(x+'A');
25 }
26 int main() {
27     int n;
28     cin >> n;
29     for (int i=1; i<=n; i++) {
30         char x, y, z;
31         cin >> x >> y >> z;
32         x = x-'A';
33         if (y == '.') {
34             a[x].left = -1;
35         } else {
36             a[x].left = y-'A';
37         }
38         if (z == '.') {
39             a[x].right = -1;
40         } else {
41             a[x].right = z-'A';
42         }
43     }
44     preorder(0);
45     cout << '\n';
46     inorder(0);
47     cout << '\n';
48     postorder(0);
49     cout << '\n';
50     return 0;
51 }
```



결과	메모리	시간	코드 길이
맞았습니다!!	1988 KB	0 ms	969 B

C++14

```
1 #include <iostream>
2 #include <algorithm>
3 #define left _left
4 #define right _right
5 using namespace std;
6 struct Node {
7     int left, right;
8     int order, depth;
9 };
10 Node a[10001];
11 int left[10001];
12 int right[10001];
13 int cnt[10001];
14 int order = 0;
15 void inorder(int node, int depth) {
16     if (node == -1) return;
17     inorder(a[node].left, depth+1);
18     a[node].order = ++order;
19     a[node].depth = depth;
20     inorder(a[node].right, depth+1);
21 }
22 int main() {
23     int n;
24     cin >> n;
25     for (int i=0; i<n; i++) {
26         int x, y, z;
27         cin >> x >> y >> z;
28         a[x].left = y;
29         a[x].right = z;
30         if (y != -1) cnt[y] += 1;
31         if (z != -1) cnt[z] += 1;
32     }
33     int root = 0;
34     for (int i=1; i<=n; i++) {
35         if (cnt[i] == 0) {
36             root = i;
37         }
38     }
39     inorder(root, 1);
40     int maxdepth = 0;
41     for (int i=1; i<=n; i++) {
42         int depth = a[i].depth;
43         int order = a[i].order;
44         if (left[depth] == 0) {
45             left[depth] = order;
46         } else {
47             left[depth] = min(left[depth], order);
48         }
49         right[depth] = max(right[depth], order);
50         maxdepth = max(maxdepth, depth);
51     }
52     int ans = 0;
53     int ans_level = 0;
54     for (int i=1; i<=maxdepth; i++) {
55         if (ans < right[i]-left[i]+1) {
56             ans = right[i]-left[i]+1;
57             ans_level = i;
58         }
59     }
60     cout << ans_level << ' ' << ans << '\n';
61     return 0;
62 }
```



결과

메모리

시간

코드 길이

맞았습니다!!

2260 KB

8 ms

1427 B

C++14

```
1 #include <cstdio>
2 #include <queue>
3 #include <vector>
4 #include <cstring>
5 using namespace std;
6 struct Edge {
7     int to;
8     int cost;
9     Edge(int to, int cost) : to(to), cost(cost) {
10    }
11 };
12 vector<Edge> a[100001];
13 bool check[100001];
14 int dist[100001];
15 void bfs(int start) {
16     memset(dist, 0, sizeof(dist));
17     memset(check, false, sizeof(check));
18     queue<int> q;
19     check[start] = true;
20     q.push(start);
21     while (!q.empty()) {
22         int x = q.front();
23         q.pop();
24         for (int i=0; i<a[x].size(); i++) {
25             Edge &e = a[x][i];
26             if (check[e.to] == false) {
27                 dist[e.to] = dist[x] + e.cost;
28                 q.push(e.to);
29                 check[e.to] = true;
30             }
31         }
32     }
33 }
34 int main() {
35     int n;
36     scanf("%d", &n);
37     for (int i=1; i<=n; i++) {
38         int x;
39         scanf("%d", &x);
40         while (true) {
41             int y, z;
42             scanf("%d", &y);
43             if (y == -1) break;
44             scanf("%d", &z);
45             a[x].push_back(Edge(y, z));
46         }
47     }
48     bfs(1);
49     int start = 1;
50     for (int i=2; i<=n; i++) {
51         if (dist[i] > dist[start]) {
52             start = i;
53         }
54     }
55     bfs(start);
56     int ans = dist[1];
57     for (int i=2; i<=n; i++) {
58         if (ans < dist[i]) {
59             ans = dist[i];
60         }
61     }
62     printf("%d\n", ans);
63     return 0;
64 }
```

결과

메모리

시간

코드 길이

맞았습니다!!

7756 KB

120 ms

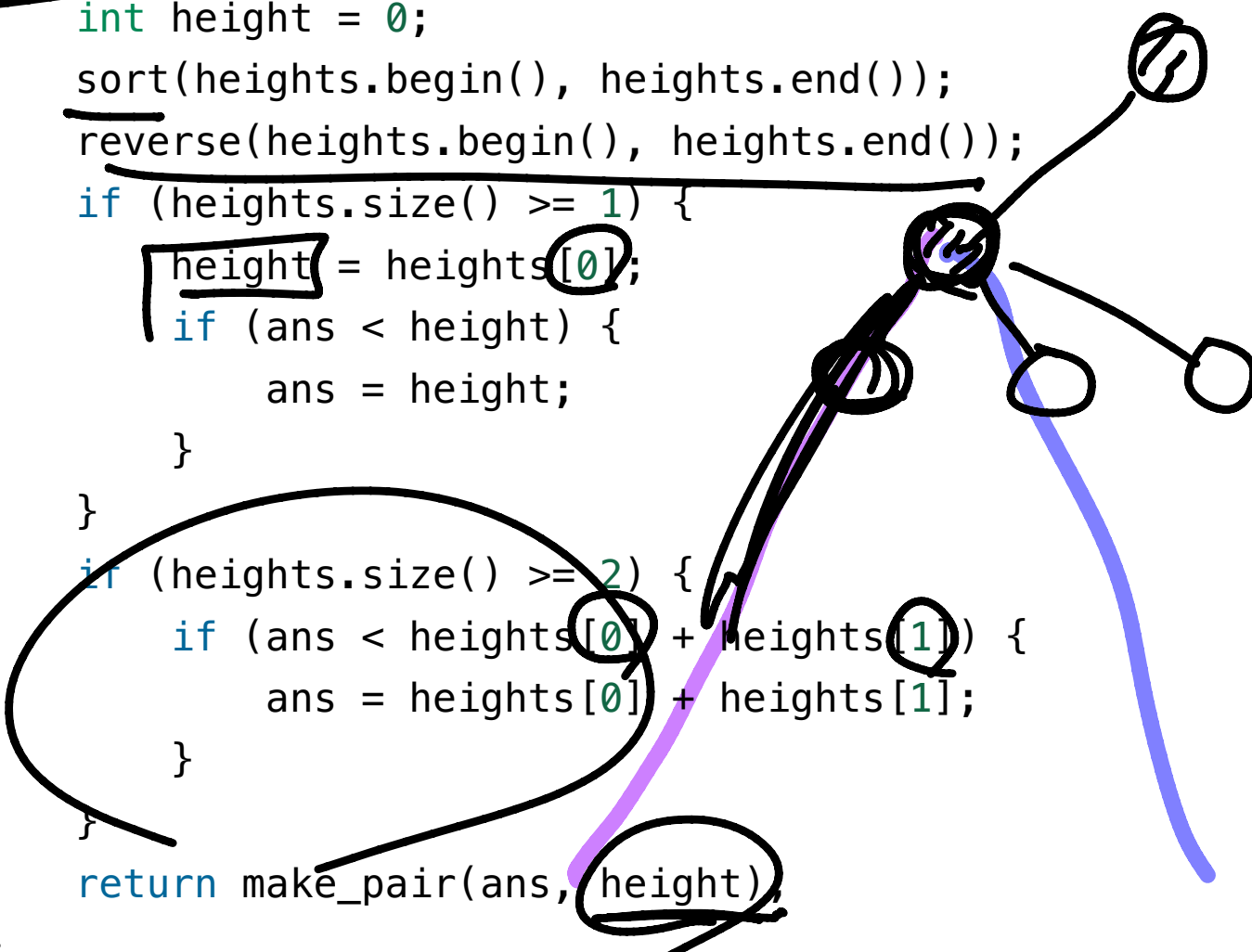
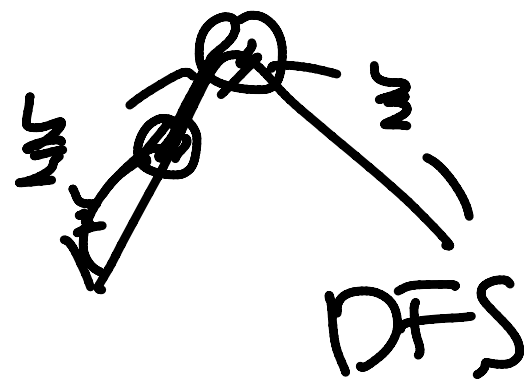
1360 B

## C++14

```

1 #include <cstdio>
2 #include <algorithm>
3 #include <iostream>
4 #include <queue>
5 #include <vector>
6 #include <cstring>
7 using namespace std;
8 struct Edge {
9     int to;
10    int cost;
11    Edge(int to, int cost) : to(to), cost(cost) {
12    }
13 };
14 vector<Edge> a[100001];
15 bool check[100001];
16 pair<int, int> dfs(int x) { // first: diameter, second: height
17     check[x] = true;
18     vector<int> heights;
19     int ans = 0;
20     for (auto &e : a[x]) {
21         int y = e.to;
22         int cost = e.cost;
23         if (check[y] == false) {
24             auto p = dfs(y);
25             if (ans < p.first) ans = p.first;
26             heights.push_back(p.second + cost);
27         }
28     }
29     int height = 0;
30     sort(heights.begin(), heights.end());
31     reverse(heights.begin(), heights.end());
32     if (heights.size() >= 1) {
33         height = heights[0];
34         if (ans < height) {
35             ans = height;
36         }
37     }
38     if (heights.size() >= 2) {
39         if (ans < heights[0] + heights[1]) {
40             ans = heights[0] + heights[1];
41         }
42     }
43     return make_pair(ans, height);
44 }
45 int main() {
46     int n;
47     scanf("%d", &n);
48     for (int i=1; i<=n; i++) {
49         int x;
50         scanf("%d", &x);
51         while (true) {
52             int y, z;
53             scanf("%d", &y);
54             if (y == -1) break;
55             scanf("%d", &z);
56             a[x].push_back(Edge(y, z));
57         }
58     }
59     auto ans = dfs(1);
60     cout << ans.first << '\n';
61     return 0;
62 }

```



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.