

WORKSHEET_6

1. Shortest Path in Binary Trees

```
#include <iostream>
using namespace std;

int main() {
    // your code goes here
    int n;
    cin >> n;
    while (n-->0) {
        int i, j, count = 0;
        cin >> i >> j;
        while (i != j) {
            if (i > j) i /= 2;
            else j /= 2;
            count++;
        }
        cout << count << endl;
    }
    return 0;
}
```

Shortest Path in Binary Trees Difficulty Rating: 1413

Statement Submissions Solution Ask a Doubt

Problem

Read problems statements in [Mandarin Chinese](#) and [Russian](#).

Consider an infinite full binary tree (each node has two children except the leaf nodes) defined as follows. For a node labelled v its left child will be labelled $2*v$ and its right child will be labelled $2*v+1$. The root is labelled as 1.

You are given N queries of the form $i\ j$. For each query, you have to print the length of the shortest path between node labelled i and node labelled j .

Next Problem

Status: ✔ Correct Answer Submission ID: [84962764](#)

Time: 0.20s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems [View another problem](#)

Upload code as file [Compile & Run](#) [Submit Code](#)

Family Tree

```
#include <iostream>
#include<bits/stdc++.h>
using namespace std;
#define ll long long

ll n, m, src, dst, cnt, r;
vector<ll> adj[100001];
ll A[100001], B[100001];
ll res = INT_MIN;

void dfs(ll u, ll ln = INT_MIN, ll nn = INT_MAX){
    ln = max(ln, A[u]);
    nn = min(nn, A[u]);
    res = max(res, ln - nn);
    for(int i: adj[u]){
        dfs(i, ln, nn);
    }
}

int main() {
    // your code goes here
    cin >> n;
    for(int i = 1; i <= n; i++) cin >> A[i];
    for(int i = 1; i <= n; i++){
        cin >> B[i];
        if(B[i] != -1) adj[B[i]].push_back(i);
    }
}
```

```

        else r = i;
    }
    dfs(r);
    cout << res;

    return 0;
}

```

The screenshot shows a web browser window displaying the CodeChef problem page for 'Family Tree'. The page is in dark mode. The left sidebar contains the problem statement, which describes a family tree with N members and asks for the maximum difference in net worth between two members where one is a descendant of the other. The right sidebar shows the submission interface, including a 'Test against Custom Input' section with the input '4 5 10 6 12 2 -1 4 2', a 'Problem Solver Badge' progress bar at 64/250, and a 'Status' section indicating a 'Correct Answer' with a submission ID of 84944832. The bottom of the page shows a Windows taskbar with various icons and system information like temperature and time.

Question 3. BLACK AND WHITE TREE

```

Language: C++14

1 #include "bits/stdc++.h"
2 using namespace std;
3 #define fast_ios_base::sync_with_stdio(false);cin.tie(0);cout.tie(0);
4 #define tt int ct;cin>>ct;while(ct-->0)
5 #define MAX 100005
6 const int mod = 998244353;
7 typedef long long ll;
8 int n,a[MAX];
9 vector<int> ad[MAX];
10 int dp[MAX][2][2];
11 void dfs(int u,int par){
12     for(auto it : ad[u]){
13         if(it != par){
14             dfs(it,u);
15         }
16     }
17     for(int i=0;i<2;i++){
18         for(int j=0;j<2;j++){
19             int odd = a[u]*i^j;
20             int cc = j;
21             int dp1[2],dp2[2];
22             memset(dp2,0x3f,sizeof(dp2));
23             dp2[0] = 0;
24             for(auto v : ad[u]){
25                 if(v != par){
26                     swap(dp1[0],dp2[0]);
27                     swap(dp1[1],dp2[1]);
28                     memset(dp2,0x3f,sizeof(dp2));
29                 }
30                 dp2[0] = min(dp2[0],dp1[0]+dp[v][cc][0]);
31                 dp2[1] = min(dp2[1],dp1[1]+dp[v][cc][0]);
32             }
33             dp2[0] = min(dp2[0],dp1[1]+dp[v][cc][1]);
34             dp2[1] = min(dp2[1],dp1[0]+dp[v][cc][1]);

```

```

35         }
36         dp[u][1][j] = dp2[odd]+j;
37     }
38     return;
39 }
40 void solve(){
41     cin>>n;
42     for(int i=0;i<n;i++){
43         cin>>a[i];
44     }
45     for(int i=0;i<n;i++){
46         ad[i].clear();
47     }
48     for(int i=0;i<n-1;i++){
49         int u,v; cin>>u>>v; u--;v--;
50         ad[u].emplace_back(v);
51         ad[v].emplace_back(u);
52     }
53     dfs(0,-1);
54     int ans = min(dp[0][0][0],dp[0][0][1]);
55     if(ans > n){
56         printf("-1\n");
57     }
58     else{
59         printf("%d\n",ans);
60     }
61     return;
62 }
63 int32_t main() {
64     fast
65     #ifndef ONLINE_JUDGE
66     freopen("input.txt","r",stdin);
67     freopen("output.txt","w",stdout);
68     #endif

```

```

70 freopen("output.txt","w",stdout);
71 #endif
72 tt{
73 solve();
74 }
75 return 0;
76 }

```

SOLUTION:

Case	Time	Memory
100	0.00	1024
1	0.00	1024
2	0.00	1024
3	0.00	1024
4	0.00	1024
5	0.00	1024
6	0.00	1024
7	0.00	1024
8	0.00	1024
9	0.00	1024
10	0.00	1024
11	0.00	1024
12	0.00	1024
13	0.00	1024
14	0.00	1024
15	0.00	1024
16	0.00	1024
17	0.00	1024
18	0.00	1024
19	0.00	1024
20	0.00	1024
21	0.00	1024
22	0.00	1024
23	0.00	1024
24	0.00	1024
25	0.00	1024
26	0.00	1024
27	0.00	1024
28	0.00	1024
29	0.00	1024
30	0.00	1024
31	0.00	1024
32	0.00	1024
33	0.00	1024
34	0.00	1024
35	0.00	1024
36	0.00	1024
37	0.00	1024
38	0.00	1024
39	0.00	1024
40	0.00	1024
41	0.00	1024
42	0.00	1024
43	0.00	1024
44	0.00	1024
45	0.00	1024
46	0.00	1024
47	0.00	1024
48	0.00	1024
49	0.00	1024
50	0.00	1024
51	0.00	1024
52	0.00	1024
53	0.00	1024
54	0.00	1024
55	0.00	1024
56	0.00	1024
57	0.00	1024
58	0.00	1024
59	0.00	1024
60	0.00	1024
61	0.00	1024
62	0.00	1024
63	0.00	1024
64	0.00	1024
65	0.00	1024
66	0.00	1024
67	0.00	1024
68	0.00	1024
69	0.00	1024
70	0.00	1024
71	0.00	1024
72	0.00	1024
73	0.00	1024
74	0.00	1024
75	0.00	1024
76	0.00	1024
77	0.00	1024
78	0.00	1024
79	0.00	1024
80	0.00	1024
81	0.00	1024
82	0.00	1024
83	0.00	1024
84	0.00	1024
85	0.00	1024
86	0.00	1024
87	0.00	1024
88	0.00	1024
89	0.00	1024
90	0.00	1024
91	0.00	1024
92	0.00	1024
93	0.00	1024
94	0.00	1024
95	0.00	1024
96	0.00	1024
97	0.00	1024
98	0.00	1024
99	0.00	1024
100	0.00	1024

Question 4. SECRET TREE

```

Language: C++14

1 #include <bits/stdc++.h>
2 #define endl '\n'
3 #define PRECISION 9
4 using namespace std;
5 using ll = long long;
6 using ld = long double;
7 #define fr first
8 #define sc second
9 using pi2 = pair<int, int>;
10 using pi1 = pair<ll, ll>;
11 #define all(v) v.begin(), v.end()
12 #define unq(v) sort( all(v) ); v.erase( unique( all(v) ), v.end() );
13
14 vector<int> adj[120]; int cnt[120];
15
16 void Main(){
17     int t; cin >> t;
18     while (t--){
19         int n; cin >> n;
20         for (int i = 2; i <= n; i++){
21             for (int j = 2; j <= n; j++){
22                 if (i==j){ continue; }
23                 cout << "2 * " << 3 << " " << i << " " << j << endl << flush;
24                 int res; cin >> res;
25                 if (res){ adj[i].push_back(j); cnt[j] += 1; }
26             }
27         }
28         for (int i = 2; i <= n; i++){ adj[i].push_back(i); cnt[i] += 1; }
29         queue<int> q; q.push(1);
30         vector<pi2> v;
31         while (!q.empty()){
32             int now = q.front(); q.pop();
33             for (int nxt : adj[now]){
34                 //cout << "EDG " << now << " " << nxt << endl << flush;
35                 for (int nxt : adj[now]){
36                     //cout << "EDG " << now << " " << nxt << endl << flush;
37                     cnt[nxt] -= 1;
38                     if (cnt[nxt] == 0){ v.push_back({now, nxt}); q.push(nxt); }
39                 }
40             }
41             cout << "!" << endl << flush;
42             for (pi2 p : v){ cout << p.fr << " " << p.sc << endl << flush; }
43             cout << flush;
44             for (int i = 1; i <= n; i++){ adj[i].clear(); cnt[i] = 0; }
45         }
46     }
47
48     int main(){
49         ios_base::sync_with_stdio(0);
50         cin.tie(0); cout.tie(0);
51         cout.setf(ios::fixed);
52         cout.precision(PRECISION);
53         Main();
54     }
55 }

```

SOLUTION:

Status: ✓ Correct Answer
Submission ID: [84941409](#)

Score:	Time:	Memory:
1	0.03s	5.4M

Question 5. BLACK AND RED VERTICES OF TREE

Language: C++14

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 using ll = long long;
4 const int mxn = 1e5+10;
5 const int mod = 1e9+7;
6 #define int ll
7 vector<int> adj[mxn];
8 int color[mxn], n, black[mxn], red[mxn];
9 ll cnt[mxn], cnt_na[mxn];
10 bool mark[mxn];
11
12 void reset() {
13     for(int i=0; i<n; i++) {
14         adj[i].clear();
15         cnt[i] = 0; cnt_na[i] = 0;
16         black[i] = red[i] = mark[i] = 0;
17     }
18 }
19
20 void dfs_rb(int v, int p) {
21     for(int x:adj[v]) if(x!=p) {
22         dfs_rb(x, v);
23         black[v] += black[x];
24         red[v] += red[x];
25     }
26     if(color[v]==1) black[v]++;
27     if(color[v]==2) red[v]++;
28 }
29
30 void dfs_mark(int v, int p) {
31     int other_black = black[0] - black[v], other_red = red[0] - red[v];
32     for(int x:adj[v]) if(x!=p) {
33         if(red[x] and other_black and !color[v]) mark[v] = 1;
34         if(black[x] and other_red and !color[v]) mark[v] = 1;
35         other_red += red[x];
36         other_black += black[x];
37     }
38     for(int x:adj[v]) if(x!=p) dfs_mark(x, v);
39 }
40
41 void dfs_cnt(int v, int p) {
42     for(int x:adj[v]) if(x!=p) {
43         dfs_cnt(x, v);
44     }
45     if(color[v]) cnt[v] = 0;
46     else {
47         cnt[v] = 1;
48         for(int x:adj[v]) if(x!=p) {
49             cnt[v] = (cnt[v] * (cnt[x] + 1)) % mod;
50         }
51     }
52 }
53
54 void dfs_cnt_na(int v, int p) {
55     for(int x:adj[v]) if(x!=p) {
56         dfs_cnt_na(x, v);
57     }
58     if(color[v]==0 and !mark[v]) {
59         cnt_na[v] = 1;
60         for(int x:adj[v]) if(x!=p) {
61             cnt_na[v] = (cnt_na[v] * (cnt_na[x] + 1)) % mod;
62         }
63     }
64     else cnt_na[v] = 0;
65 }
66
67 signed main() {
68     ios_base::sync_with_stdio(0); cin.tie(0);
69
70     int tc; cin>>tc;
```

```

70 int tc; cin>>tc;
71 while(tc--) {
72     cin>>n;
73     reset();
74     for(int i=0; i<n-1; i++) {
75         int u, v; cin>>u>>v;
76         v--; u--;
77         adj[u].push_back(v);
78         adj[v].push_back(u);
79     }
80     for(int i=0; i<n; i++) cin>>color[i];
81     dfs_rb(0, 0);
82     // cout << "black: "; for(int i=0; i<n; i++) cout << black[i] << " "; cout << '\n';
83     // cout << "red: "; for(int i=0; i<n; i++) cout << red[i] << " "; cout << '\n';
84     dfs_mark(0, 0);
85     // cout << "mark: "; for(int i=0; i<n; i++) cout << mark[i] << " "; cout << '\n';
86     dfs_cnt(0, 0);
87     //cout << "cnt: "; for(int i=0; i<n; i++) cout << cnt[i] << " "; cout << '\n';
88     dfs_cnt_na(0, 0);
89
90     ll sumall = 0;
91     for(int i=0; i<n; i++) {
92         sumall+=cnt[i];
93         sumall%=mod;
94     }
95     ll sumnotmark = 0;
96     for(int i=0; i<n; i++) {
97         sumnotmark+=cnt_na[i];
98         sumnotmark%=mod;
99     }
100     ll ans = (sumall - sumnotmark + mod)%mod;
101     cout << ans << '\n';
102 }
103 return 0;
104 }

```

SOLUTION:

Status: ✓ Correct Answer		Submission ID: 84945342
Time:	Memory:	
0.42s	23.2M	

Question 6. COMMON ANCESTORS

Language: C++14

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 const int MX = 1e6 + 10;
6
7 int seg[4 * MX], lazy[4 * MX], beg[MX], fin[MX], dep[MX], cnt;
8 vector<int> adj[2][MX];
9
10 void updateNode(int idx, int v) {
11     seg[idx] += v;
12     lazy[idx] += v;
13
14     return;
15 }
16
17 void shift(int idx, int st, int ed) {
18     int lft = 2 * idx, rgt = lft + 1;
19
20     if (lazy[idx]) {
21         updateNode(lft, lazy[idx]);
22         updateNode(rgt, lazy[idx]);
23
24         lazy[idx] = 0;
25     }
26
27     return;
28 }
29
30 void update(int s, int e, int v, int idx = 1, int st = 0, int ed = cnt - 1) {
31     if (s > e || e < st || s > ed) return;
32
33     if (s == st && e == ed) {
34         updateNode(idx, v);
35
36         return;
37     }
38
39     int lft = 2 * idx, rgt = lft + 1, mid = (st + ed) / 2;
40
41     shift(idx, st, ed);
42
43     update(s, min(e, mid), v, lft, st, mid), update(max(s, mid + 1), e, v, rgt, mid + 1, ed);
44
45     seg[idx] = max(seg[lft], seg[rgt]);
46
47     return;
48 }
49
50 void dfs0(int u, int d) {
51     beg[u] = cnt++;
52     dep[u] = d;
53
54     for (auto v : adj[0][u]) dfs0(v, d + 1);
55
56     fin[u] = cnt - 1;
57
58     return;
59 }
60
61 void dfs1(int u, int d, int ans) {
62     if (dep[u] == d) update(beg[u], fin[u], 1);
63
64     ans = max(ans, seg[1]);
65
66     for (auto v : adj[1][u]) {
67         dfs1(v, d + 1, ans);
68     }
69
70     if (dep[u] == d) update(beg[u], fin[u], -1);
71
72     return;
```

```

72     return;
73 }
74
75 int main() {
76     ios::sync_with_stdio(false);
77     cin.tie(0);
78     cout.tie(0);
79
80     int t;
81     cin >> t;
82
83     while (t--) {
84         cnt = 0;
85
86         int n;
87         cin >> n;
88
89         for (int i = 0; i <= 4 * n; i++) {
90             seg[i] = lazy[i] = 0;
91
92             if (i <= n) {
93                 adj[0][i].clear();
94                 adj[1][i].clear();
95             }
96         }
97
98         for (int j = 0; j < 2; j++) {
99             for (int i = 1; i <= n; i++) {
100                 int p;
101                 cin >> p;
102
103                 adj[j][p != -1 ? p : 0].push_back(i);
104             }
105         }
106
107         dfs0(0, 0);
108
109         int ans = 0;
110
111         dfs1(0, 0, ans);
112
113         cout << ans - 1 << endl;
114     }
115
116     return 0;
117 }

```

SOLUTION:

Status: ✓ Correct Answer		Submission ID: 84946415
Time: 1.23s	Memory: 185.2M	

Question 7. SUBTREE REMOVAL

```

Language: C++14

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  #define ll long long
5  const ll Nn = 1e5 + 7;
6
7  ll P[Nn];
8  vector <int> Adj[Nn];
9  ll X;
10 void DFS(int u, int p) {
11     ll t = P[u];
12     for (int v : Adj[u]) if (v != p){
13         DFS(v, u);
14         t += max(P[v], -X);
15     }
16     P[u] = max(t, -X);
17 }
18
19 int main()
20 {
21     int t;
22     cin >> t;
23
24     while (t--) {
25         ll N;
26         cin >> N >> X;
27         for (int i = 1; i <= N; ++i){
28             cin >> P[i];
29             Adj[i].clear();
30         }
31
32         for (int i = 1; i < N; ++i) {
33             int u, v;
34             cin >> u >> v;
35             Adj[u].push_back(v);
36             Adj[v].push_back(u);
37         }
38
39         DFS(1, 0);
40
41         cout << P[1] << "\n";
42     }
43     return 0;
44 }

```

SOLUTION:

Status: ✓ Correct Answer			Submission ID: sub84786
Score: 100	Time: 0.79s	Memory: 16.5M	
Sub-Test	Test #	Result (ms)	
1	1	AC (0.00470s)	
1	2	AC (0.00490s)	
1	3	AC (0.00485s)	
1	4	AC (0.00512s)	
1	5	AC (0.00509s)	
Subtask Score: 50.00%		Result: AC	
2	6	AC (0.01076s)	
2	7	AC (0.00271s)	
2	8	AC (0.00708s)	
2	9	AC (0.00214s)	
2	10	AC (0.00277s)	
Subtask Score: 70.00%		Result: AC	
Total Score = 100.00%			

Question 8. ALTERNATING DIAMETER

```
Language: C++14

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define int long long int
4
5  int32_t main()
6  {
7      int t;
8      cin >> t;
9      while (t--)
10     {
11         int b, w;
12         cin >> b >> w;
13         vector<char> c('B', 'W');
14         if (b < w)
15         {
16             swap(b, w);
17             swap(c[0], c[1]);
18         }
19         if (w == 0 && b > 1)
20         {
21             cout << -1 << endl;
22             continue;
23         }
24         int a = b + w;
25         if (a >= 1)
26             cout << c[0];
27         if (a >= 2)
28             cout << c[1];
29         if (a >= 3)
30             cout << c[0];
31         b -= 2;
32         w--;
33         for (int i = 0; i < b; i++)
34             cout << c[0];
35         for (int i = 0; i < w; i++)
36             cout << c[1];
37         cout << endl;
38         for (int i = 1; i <= a; i++)
39         {
40             if (i != 2 && a > 1)
41                 cout << i << " " << 2 << endl;
42         }
43     }
44 }
```

SOLUTION:

Status: ✓ Correct Answer			Submission ID: 84947044		
Score:	Time:	Memory:			
100	0.15s	5.3M			
Sub-Task		Task #	Result (Time)		
1		1	AC (0.00009s)		
1		2	AC (0.075577)		
1		3	AC (0.078191)		
1		4	AC (0.147099)		
1		5	AC (0.073408)		
Subtask Score: 100.00%			Result - AC		
Total Score = 100.00%					