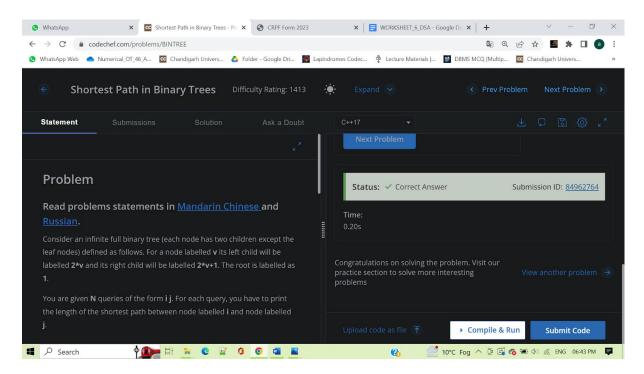
WORKSHEET_6

1. Shortest Path in Binary Trees

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



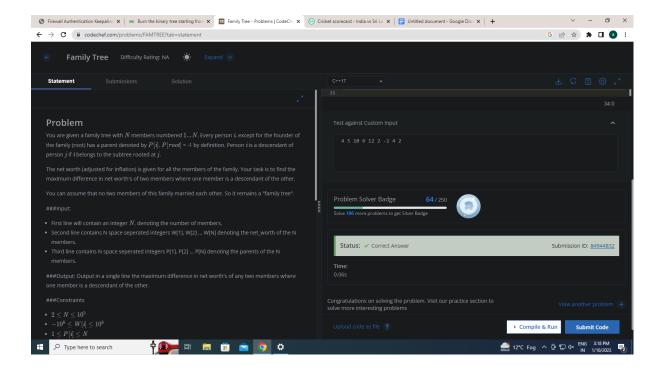


Family Tree

```
#include <iostream>
#include<bits/stdc++.h>
using namespace std;
#define II long long
Il n, m, src, dst, cnt, r;
vector<ll> adj[100001];
II A[100001], B[100001];
II res = INT_MIN;
void dfs(II u, II In = INT_MIN, II 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 \le 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;
}</pre>
```



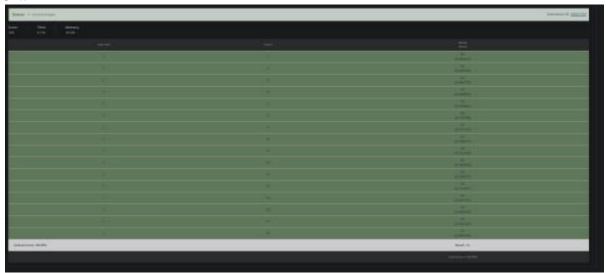
Question 3. BLACK AND WHITE TREE



```
36 dp[u][i][j] = dp2[odd]+j;
37 } }
38 neturn;
39 }
39 void solve(){
41 cin>n;
42 for(int i-0;icn;i++){
43 cin>>a[i];
44 }
45
46 for(int i-0;icn-1;i++){
47 ad[i].clear();
48 }
49 for(int i-0;icn-1;i++){
50 int u,v; cin>u>v>v;
51 ad[u].emplace back(v);
52 ad[v].emplace back(v);
53 }
54 dfs(0,-i);
56 int ans = min(dp[0][0][d],dp[0][0][i]];
57 if(ans > n){
58 printf("+ln");
59 }
61 else{
62 printf("*dhn",ans);
63 }
64 neturn;
65 }
66 int32 t main() {
67 fast
68 gfeopen("input.txt","",stdin);
78 froopen("output.txt","u",stdout);
78 vandif
```



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



Question 4. SECRET TREE



```
Language: C++14
           #include <bits/stdc++.h>
#define endl '\n'
           #define PRECISION 9
         4 using mamempace std;
5 using 11 = long long;
6 using 1d = long double;
7 modfine for first
        void Main()[
int t; cln >> t;
               If (res){ adj[1].push_back(j); cnt[j] += 1; }
                   for (int i=2; i < n; i\leftrightarrow){ adj[i].push_back(i); cnt[i] \leftrightarrow i; } queue<int> q; q.push(i);
                   vector(pi2) v;
                   while (!q.empty()){
                      int now = q.front(); q.pop();
for (int nxt : adj[now]){
    //cout << "EDG" << now <</pre>
               for (int nxt : adj[now]){
    //cout << "EDG" << now << ' ' << nxt << endl << flush;</pre>
                    cnt[nxt] -= 1;
                    if (cnt[nxt] == 0){ v.push_back({now, nxt}); q.push(nxt); }
          cout << "!" << endl << flush;
          for (pi2 p : v){ cout << p.fr << ' ' << p.sc << endl << flush; }
          cout << flush;</pre>
          for (int i = 1; i <= n; i++){ adj[i].clear(); cnt[i] = 0; }
int main(){
     ios_base::sync_with_stdio(0);
     cin.tie(0); cout.tie(0);
     cout.setf(ios::fixed);
     cout.precision(PRECISION);
     Main();
```

37 38

39

40

46

47

48

49

50



Question 5. BLACK AND RED VERTICES OF TREE



```
Language: C++14
          using namespace std;
        using 11 - long long;
const int man - 105410;
     5 const int mod = 109+7;
    7 vector(int) adj[mxn];
8 int color[mxn], n, black[mxn], red[mxn];
9 11 cnt[mxn], cnt_na[mxn];
   10 bool mark[mun];
   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;
   28 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];
         if(color[v]--1) black[v]+-1;
if(color[v]--2) red[v]+-1;
   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;
34 if(black[x] and other_red and !color[v]) mark[v] = 1;
35 other_red+=red[x];
36 other_black+=black[x];
 38 for(int x:adj[v]) if(x!-p) dfs_mark(x, v);
48
41 void dfs_cnt(int v, int p) {
42 for(int x:adj[v]) if(x!-p) {
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;
54 void dfs_cnt_na(int v, int p) {
55 for(int x:adj[v]) if(x!-p) {
56 dfs_cnt_na(x, v);
58 if(color[v]-+0 and lmark[v]) (
59 cnt_na[v] = 1;
60 for(int x:adj[v]) if(xl=p) {
61 cnt_na[v] = (cnt_na[v] = (cnt_na[x] + 1)) % mod;
64 else cnt_na[v] = 0;
67 signed main() (
68 ios_base::sync_with_stdio(0); cin.tie(0);
```



```
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);
 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);
88 dfs_cnt_na(0, 0);
 89
90 ll sumall = 0;
91 for(int i=0; i<n; i++) {
92 sumall+=cnt[i];
     sumall%=mod;
94
 95 11 sumnotmark = 0;
96 for(int i=0; i<n; i++) {
97 sumnotmark+=cnt_na[i];
98 sumnotmark%=mod;
100 ll ans = (sumall - sumnotmark + mod)%mod;
101 cout << ans << '\n';
103
104
```



Question 6. COMMON ANCESTORS



```
int ift * 2 * idx, rgt = lft * 1, mid * (st + ed) / 3;
int ift * 2 * idx, rgt = lft * 1, mid * (st + ed) / 3;

shift(idx, st, ed);

update(s, min(e, mid), v, lft, st, mid), update(max(s, mid * 1), e, v, rgt, mid * 1, ed);

seg[idx] = max(seg[ift], seg[rgt]);

rutuum;

veid dfsd(int u, int d) {
    beg[u] = cnt * 1;
    dep[u] = ent * 1;
    inturn;

inturn
```



```
return;
}

return;
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    int t;
    cin > t;

    while (t-.) {
        cnt = 0;

        int n;
        cin > n;

        for (int i = 0; i <= 4 * n; i++) {
            seg[i] = lazy[i] = 0;

        if (i <= n) {
            adj[0][i].clear();
            yd adj[0][i].clear();
            yd
```



Question 7. SUBTREE REMOVAL



Statute: - / Corest Andrew Summer Co			Submission ID. Substition
kare: Nese 00 0.7%	Memory 163M		
			No.6 Street
			MAN COLUMN TO THE PARTY OF THE
			nation.
			K A THINKS
		4	e e e e e e e e e e e e e e e e e e e
		4.0	
blank line 2.3%			Natural AC
	9	45	10.0107E2
	9	4	E DANTA
	4		C (1)558
		(8)	2504
	- 32		p.Ss
Substant Score: 76.00%			Name AC
			Plant State Million



Question 8. ALTERNATING DIAMETER

SOLUTION:

