



# **WORKSHEET 6**

Student Name: Manpreet Kaur UID: 20BCS2212

**DOMAIN CAMP:** 03-01-2023 to 14-01-2023 **Section/Group:** DWWC-43

**Subject Name:** IT Skills (DSA)

## **Question 1. FAMILY TREE**

```
Language: C++14
       #include <bits/stdc++.h>
       using namespace std;
      #define 11 long long
   6 #define ld long double
       #define pb push_back
   8 #define pf push_front
 9 #define mp make_pair
10 #define all(v) v.begin(), v.end()
11 #define test() int t; cin >> t; while(t--)
 12 #define nl cout << endl
14 ll n, m, src, dst, cnt, r;
15 vector(ll> adj[100001];
16 ll A[100001], B[100001];
17 ll ln[100001], nn[100001];
18 ll res - 123456789;
       void dfs(11 u){
               for(int i: adj[u]){
    ln[i] = max(ln[u], A[i]);
    nn[i] = min(nn[u], A[i]);
    dfs(i);
 21
22
23
24
25
26
27
28 main(){
29 ios
                ios_base::sync_with_stdio(0);
               cin.tie(0); cout.tie(0);
if(fopen("inp.inp", "r")){
   freopen("inp.inp", "r", stdin);
   freopen("out.out", "w", stdout);
 30
31
```

```
35
36
         cin >> n;
         for(int i = 1; i \leftarrow n; i++) cin >> A[i];
38 -
         for(int i = 1; i \leftarrow n; i++){
             cin >> B[i];
40
              if(B[i] != -1) adj[B[i]].pb(i);
              else r = i;
41
42
         ln[r] = -123456789;
nn[r] = 123456789;
44
         dfs(r);
46
         for(int i = 1; i \le n; i++) res = max(res, ln[i] - nn[i]);
         cout << res;
48 }
```







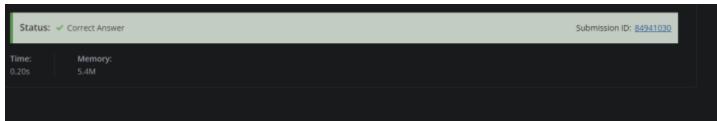
```
Status: ✓ Correct Answer

Time: Memory:
0.03s 10.7M
```

# **Question 2. SHORTEST PATH IN BINARY TREES**

```
Language: C++14

1  #include <iostream>
2  using namespace std;
3
4- int main() {
5    int t;
6    cin>t;
7- while(t--) {
8    int L, R;
9    cin>slx>R;
10    int cnt=0;
11- while(L!=R) {
12-    if(L>R) {
13         L=L/2;;
14    }
15-    else {
16         R=R/2;
17    }
18    cnt++;
19   }
20   cout<<cnt<<end1;
21  }
22  return 0;
23 }</pre>
```









### **Question 3. BLACK AND WHITE TREE**

```
Language: C++14
  using namenpace std;
define fast ios_base::sync_with_stdio(false);cin.tie(0);cout.tie(0);
  4 #define tt int ct;cin>>ct;while(ct--)
  5 #define MAX 100005
  6 const int mod = 998244353;
7 typedef long long 11;
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);
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[8] = 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));
30 dp2[0] = min(dp2[0],dp1[0]+dp[v][cc][0]);
31 dp2[1] = min(dp2[1],dp1[1]+dp[v][cc][0]);
 33 dp2[0] = min(dp2[0],dp1[1]+dp[v][cc][1]);
 34 dp2[1] - min(dp2[1],dp1[0]+dp[v][cc][1]);
```

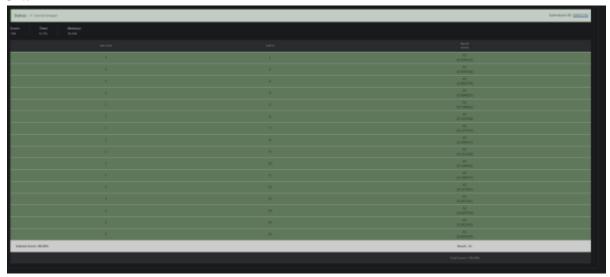
```
35 } }
36 dp[u][1][j] = dp2[odd]+j;
37 ] }
38 return;
39 }
40 void solve(){
41 cin>>n;
42 for(int i=0;i<n;i++){
43 cin>>a[i];
44 }
45
46 for(int i=0;i<n;i++){
47 ad[i].clear();
48 }
49 for(int i=0;i<n-1;i++){
50 int u,v; cin>u>v; u-;v-;
51 ad[u].emplace_back(v);
52 ad[v].emplace_back(v);
53 }
54
55 int ans = win(dp[e][0][0],dp[e][e][1]);
56 int ans = win(dp[e][0][0],dp[e][e][1]);
57 int ans = win(dp[e][0][0],dp[e][e][1]);
58 if(ans > n)(
50 printf("i\n");
50 printf("i\n");
60 intid = finder ONLINE_JUDGE
69 freopen("input.txt","",stdout);
71 #endif
```







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



## **Question 4. SECRET TREE**







```
(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); }
39
             cout << "!" << endl << flush;
             for (pi2 p : v){ cout << p.fr << ' ' << p.sc << endl << flush; }
40
             cout << flush;
             for (int i = 1; i <= n; i++){ adj[i].clear(); cnt[i] = 0; }
43
   int main(){
46
47
         ios base::sync with stdio(0);
         cin.tie(0); cout.tie(0);
cout.setf(ios::fixed);
48
49
         cout.precision(PRECISION);
50
         Main();
```



## **Question 5. BLACK AND RED VERTICES OF TREE**





```
other_red+-red[x];
36 other_black+-black[x];
      for(int x:adj[v]) if(x!-p) dfs_mark(x, v);
41 void dfs_cnt(int v, int p) {
42 for(int x:adj[v]) if(x!-p) {
43 dfs_cnt(x, v);
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 !mark[v]) (
59 cnt_na[v] = 1;
60 for(int x:adj[v]) if(x!=p) {
     cnt_na[v] = (cnt_na[v] * (cnt_na[x] + 1)) % mod;
    else cnt_na[v] = 0;
67 signed main() {
68 ios_base::sync_with_stdio(0); cin.tie(0);
69
```

```
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);
 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);
90 ll sumall = 0;
91 for(int i=0; i<n; i++) {
92 sumall+=cnt[i];
93 sumall%=mod;
 94
 95 11 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
104
```







## **Question 6. COMMON ANCESTORS**

```
int lft + 2 * idx, rgt - lft + 1, mid + (st + ed) / 3;
int lft + 2 * idx, rgt - lft + 1, mid + (st + ed) / 3;
int lft + 2 * idx, rgt - lft + 1, mid + (st + ed) / 3;
int lft (ldx, st, ed);
update(s, min(e, mid), v, lft, st, mid), update(max(s, mid + 1), e, v, rgt, mid + 1, ed);
is seg[idx] - max(seg[ift], seg[rgt]);
ruturn;
}
uvoid dfs0(int u, int d) {
leg[u] - cnt++;
dep[u] - d;
if (ruto v : adj[n][u]) dfs0(v, d + 1);
if in[u] - cnt - 1;
if ruturn;
}
uvoid dfs1(int u, int d, int Aans) {
if (dep[u] - v) update(beg[u], fin[u], 1);
if ans - max(ans, seg[1]);
if (dep[u] - d) update(beg[u], fin[u], -1);
if (dep[u] - d) update(beg[u], fin[u], -1);
if continue.
```







```
return;
}
int main() {
ios::sync_with_stdio(false);
cin.tie(0);

sector(int);

se
```

```
Status: ✓ Correct Answer

Submission ID: <u>84946415</u>

Time: Memory:
1.23s 185.2M
```

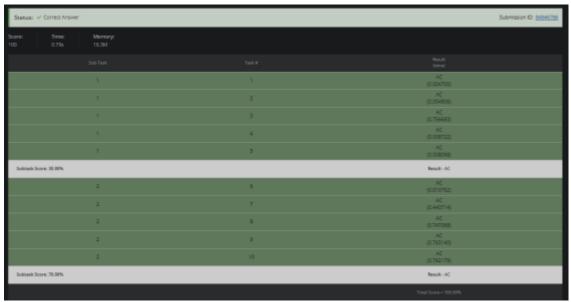






## **Question 7. SUBTREE REMOVAL**

```
Language: C++14
    #include <bits/stdc++.h>
using namespace std;
    #define 11 long long
const 11 Nn = 1e5 + 7;
13
14
          }
P[u] = max(t, -X);
    int main()
20
21
          int t;
cin >> t;
22
23
24
25
              27
28
29
               int u, v;
cin >> u >> v;
Adj[u].push_back(v);
Adj[v].push_back(u);
33
34
35
36
37
38
39
40
41
               DFS(1, 0);
               cout << P[1] << "\n";
           return 0;
43
44
```









## **Question 8. ALTERNATING DIAMETER**



