

## WORKSHEET 6

**Student Name:** Manpreet Kaur

**DOMAIN CAMP:** 03-01-2023 to 14-01-2023

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

**UID:** 20BCS2212

**Section/Group:** DWWC-43

### Question 1. FAMILY TREE

```
Language: C++14

1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  #define ll long long
6  #define ld long double
7  #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
13
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;
19
20 void dfs(ll u){
21     for(int i: adj[u]){
22         ln[i] = max(ln[u], A[i]);
23         nn[i] = min(nn[u], A[i]);
24         dfs(i);
25     }
26 }
27
28 main(){
29     ios_base::sync_with_stdio(0);
30     cin.tie(0); cout.tie(0);
31     if(fopen("inp.inp", "r")){
32         freopen("inp.inp", "r", stdin);
33         freopen("out.out", "w", stdout);
34     }
35
36     cin >> n;
37     for(int i = 1; i <= n; i++) cin >> A[i];
38     for(int i = 1; i <= n; i++){
39         cin >> B[i];
40         if(B[i] != -1) adj[B[i]].pb(i);
41         else r = i;
42     }
43     ln[r] = -123456789;
44     nn[r] = 123456789;
45     dfs(r);
46     for(int i = 1; i <= n; i++) res = max(res, ln[i] - nn[i]);
47     cout << res;
48 }
```

**SOLUTION:**

Status:  Correct Answer

Time:  
0.03s


Memory:  
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>>L>>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<<endl;
21     }
22     return 0;
23 }
```

**SOLUTION:**

Status:  Correct Answer

Submission ID: [84941030](#)

Time:  
0.20s

Memory:  
5.4M

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

```

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

**SOLUTION:**

Statistics: 1/1 Correct Answer				Submission ID: 100
--------------------------------	--	--	--	--

### Question 4. SECRET TREE

```

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 pi2 = 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 << " ? " << i << " " << j << " " << 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 << " ? " << now << " " << nxt << endl << flush;

```

```

33-         for (int nxt : adj[now]){
34-             //cout << "EDG " << now << ' ' << nxt << endl << flush;
35-             cnt[nxt] -= 1;
36-             if (cnt[nxt] == 0){ v.push_back({now, nxt}); q.push(nxt); }
37-         }
38-     }
39-     cout << "!" << endl << flush;
40-     for (pi2 p : v){ cout << p.fr << ' ' << p.sc << endl << flush; }
41-     cout << flush;
42-     for (int i = 1; i <= n; i++){ adj[i].clear(); cnt[i] = 0; }
43- }
44- }
45-
46- int main(){
47-     ios_base::sync_with_stdio(0);
48-     cin.tie(0); cout.tie(0);
49-     cout.setf(ios::fixed);
50-     cout.precision(PRECISION);
51-     Main();
52- }

```

0:0

**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;

```



```

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: 84945942

Time: 0.42s

Memory: 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
15 void shift(int idx, int st, int ed) {
16     int lft = 2 * idx, rgt = lft + 1;
17
18     if (lazy[idx]) {
19         updateNode(lft, lazy[idx]);
20         updateNode(rgt, lazy[idx]);
21         lazy[idx] = 0;
22     }
23 }
24
25 void update(int s, int e, int v, int idx = 1, int st = 0, int ed = cnt - 1) {
26     if (s > e || e < st || s > ed) return;
27
28     if (s == st && e == ed) {
29         updateNode(idx, v);
30     }
31
32     return;
33 }
34
35 void dfs0(int u, int d) {
36     beg[u] = cnt++;
37     dep[u] = d;
38
39     for (auto v : adj[0][u]) dfs0(v, d + 1);
40
41     fin[u] = cnt - 1;
42     return;
43 }
44
45 void dfs1(int u, int d, int ans) {
46     if (dep[u] == d) update(beg[u], fin[u], 1);
47
48     ans = max(ans, seg[1]);
49
50     for (auto v : adj[1][u]) {
51         dfs1(v, d + 1, ans);
52     }
53
54     if (dep[u] == d) update(beg[u], fin[u], -1);
55 }
56
57 int main() {
58     int n, m;
59     cin >> n >> m;
60
61     for (int i = 1; i <= n; i++) {
62         int p;
63         cin >> p;
64         adj[0][i].push_back(p);
65     }
66
67     for (int i = 1; i <= m; i++) {
68         int u, v;
69         cin >> u >> v;
70         adj[1][u].push_back(v);
71         adj[1][v].push_back(u);
72     }
73
74     dfs0(1, 0);
75     dfs1(1, 0, 0);
76
77     cout << seg[1] << endl;
78 }
```

```
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 }
```

0:0

## 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: [A9445796](#)

Score:	Time:	Memory:
100	0.79s	16.3M

Sub-Task	Task #	Result (Time)
1	1	AC (0.004700)
1	2	AC (0.004900)
1	3	AC (0.756683)
1	4	AC (0.008722)
1	5	AC (0.008098)
Subtask Score: 10.00%		Result: AC
2	6	AC (0.010752)
2	7	AC (0.440714)
2	8	AC (0.747068)
2	9	AC (0.763140)
2	10	AC (0.762179)
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: 100

Time: 0.15s

Memory: 5.3M

Sub-Task	Task #	Result (time)
1	1	AC (0.005096)
1	2	AC (0.075577)
1	3	AC (0.076191)
1	4	AC (0.147099)
1	5	AC (0.073409)
Subtask Score: 100.00%		Result - AC
Total Score = 100.00%		