

Problem-3

Aim of the Experiment :

You are given a tree with N vertices (numbered 1 through N) and a sequence of integers A_1, A_2, \dots, A_N . You may choose an arbitrary permutation p_1, p_2, \dots, p_N of the integers 1 through N . For each valid i , you should assign the value A_{p_i} to vertex i .

In this process, you obtain a sequence of vertices v_1, v_2, \dots, v_K — the initial vertex and the vertices in which you were after each operation. Your score is

$\sum_{i=1}^K A_{p_{v_i}}$. What is the maximum possible value of this score?

1. Problem Description :

A tree is given having N vertices and a sequence of integers A_1, \dots, A_N . We take arbitrary permutation of integers 1 to N and assign the value A_{p_i} to the vertex i . Then we will choose a vertex and move forward and move backward $(K-1)$ times. We will obtain a sequence of vertices v_1, \dots, v_K i.e. the initial vertex and the vertices in which you were after each operation. Then we will calculate the score using the formula and find its maximum possible value for this score.

2. Algorithm :

function dfs($x, p = -1$):

Step 1: Define a DFS function that traverses the tree, calculates the distance from the root to each node ($d[x]$), and finds the minimum sum of distances (mn).

Step 2: If the current node is a leaf (size of its adjacency list is 1) and it's not the root, update the minimum sum (mn).

Step 3: Recursively explore the adjacent nodes, keeping track of the two smallest distances (m_1 and m_2).

function solve():

Step 1: Read the number of nodes n and the target sum k .

NAME: ASHISH KUMAR

UID: 23MAI10008

Course Name: ADSA Lab

Course Code: 23CSH-622

Step 2: Read the array a representing the values associated with each node.

Step 3: Build the tree by reading the edges between nodes and populating the adjacency list.

Step 4: Find the root of the tree by identifying the node with only one adjacency (leaf node).

Step 5: Initialize the minimum distance variable (mn) and calculate distances using DFS.

Step 6: Sort the array a in descending order.

Step 7: Calculate the sum s of the top mn values in the sorted array.

Step 8: Adjust k based on the relationship between k and mn.

Step 9: Calculate the final answer (ans) by summing the appropriate values from the array.

Step 10: Output the result..

3. Source Code for Experiment :

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
using ld = long double;
#define ft first
#define sd second

constexpr ll N = ll(3e5) + 5;
constexpr int MOD = int(1e9) + 7;
constexpr int inf = 0x3f3f3f3f;

int a[N], d[N], mn;
vector<int> v[N];

int dfs(int x, int p = -1){
    if(v[x].size() == 1 && p != -1){
        mn = min(mn, d[x]);
        return 1;
    }
    int m1 = inf, m2 = inf;
```

Course Name: ADSA Lab

Course Code: 23CSH-622

```
for(auto& i : v[x]){
    if(i == p) continue;
    d[i] = 1 + d[x];
    int k = dfs(i, x);
    if(k < m1){
        m2 = m1;
        m1 = k;
    }
    else if(k < m2) m2 = k;
}
if(m2 != inf) mn = min(mn, m1 + m2);
return m1 + 1;
}
```

```
void solve(){
    int n, k;
    cin >> n >> k;

    for(int i = 0; i < n; i++) cin >> a[i];
    for(int i = 1; i < n; i++){
        int x, y;
        cin >> x >> y;

        v[x].push_back(y);
        v[y].push_back(x);
    }
    int r = 1;
    for(int i = 1; i <= n; i++){
        if(v[i].size() == 1) r = i;
    }
    mn = inf;
    d[r] = 0;
    dfs(r);
    mn++;

    sort(a, a + n, greater<int>());
```

Course Name: ADSA Lab

Course Code: 23CSH-622

```
ll ans = 0, s = 0;

for(int i = 0; i < mn; i++) s += a[i];
if((k / mn) & 1){
    ans = (k / mn - 1) * s;
    k = k % mn + mn;
}
else{
    ans = k / mn * s;
    k = k % mn;
}
if(k < 3){
    ans += a[0] * (k >= 1);
    ans += a[1] * (k == 2);
}
else for(int i = 0; i < mn && k > 0; i++){
    ans += min(k, 2) * a[i];
    k -= 2;
}
cout << ans << '\n';
for(int i = 1; i <= n; i++) v[i].clear();
}

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

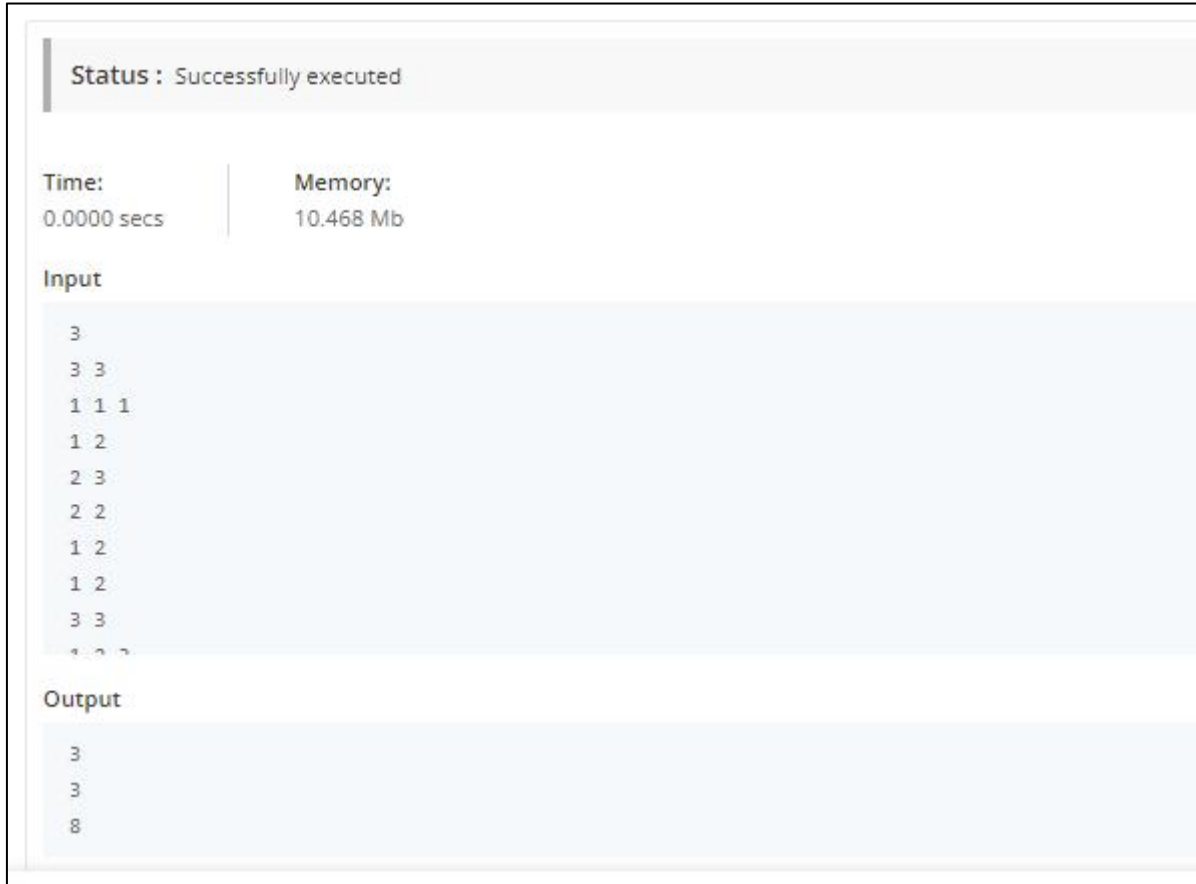
    int tt = 1;
    cin >> tt;

    for(int i = 0; i < tt; i++) {
        solve();
    }
}
```

Course Name: ADSA Lab

Course Code: 23CSH-622

4. Result/Output :



The screenshot shows a C++ IDE interface with the following details:

- Status:** Successfully executed
- Time:** 0.0000 secs
- Memory:** 10.468 Mb
- Input:**

```
3
3 3
1 1 1
1 2
2 3
2 2
1 2
1 2
3 3
1 2 2
```
- Output:**

```
3
3
8
```

Learning outcomes (What I have learnt):

1. I learnt about the linked list data structure in C++.
2. I learnt about the fast and slow pointers in linked list.
3. I learnt about how to find the middle node in linked list.
4. I learnt about how to reverse the linked list.
5. I learnt about how to check palindrome linked list.