Problem

Submissions

Discussion Coming Soon

You will be given a tree with the length of each edge equal to $1$. The nodes of the tree are numbered from 1 to N. Print the node $x$ that will minimize the sum of $f(x, a)$ + $f(x, b)$ + $f(x, c)$ for the given nodes $a$, $b$ and $c$.

$f(a, b)$ defined as the length of the simple path between two nodes on a tree.

### Input Format

$N$, the number of nodes in the tree will be given in first line. Next $N - 1$ lines will contain 2 integers and describe the edges of the tree in form $k_i$ and $l_i$. Last line will contain 3 integers $a$, $b$, and $c$, number of nodes given in the statement.

### Output Format

The number of node $x$ that minimizes the sum of $f(x, a)$ + $f(x, b)$ + $f(x, c)$.

If there are multiple answers, print the smallest one.

### Constraints

$1 \leq N \leq 10^5$

$a_i \neq b_i, a_i \neq c_i, b_i \neq c_i$

$1 \leq a_i, , b_i, c_i \leq N$

It is guaranteed that given edges form a tree.

Sample Input 1

```
7
1 2
5 2
1 4
2 3
6 2
7 2
5 3 6
```

Sample Output 1

```
2
```

Sample Input 2

```
7
1 2
5 3
4 1
7 5
3 2
1 6
7 5 6
```

Sample Output 2

```
5
```

C++ (GCC 9.2.0) Bright **Memory Limit (kB) :** 256000 **Time Limit (s) :** 1

```cpp
#include <bits/stdc++.h>

const int MAX = 100005;
using namespace std;
bool visited[MAX];
vector<int> adj[MAX];
int dist[MAX];

void    BFS(int src){
    queue<int> q;
    q.push(src);
    visited[src] = true;
    dist[src] = 0;
    while(!q.empty()) {
        int u = q.front();
        q.pop();
        for(auto v: adj[u]){
            if(!visited[v]){
                visited[v] = true;
                q.push(v);
                dist[v] = dist[u] + 1;
            }
        }
    }
}

int main() {
    int n;
    cin >> n;
    for(int i = 0; i < n - 1; i++){
        int u, v;
```

Upload File  Test against custom test case  Run Code  Submit

Sample Test Case 0

Sample Test Case 1

## Accepted

**Input(stdin)**

```
7
1 2
5 2
1 4
2 3
6 2
7 2
5 3 6
```

**Output(stdin)**

```
2
```

**Expected Output**

```
2
```