

MARSUPİLAMİLER - ÇÖZÜM

Çözüm için işe yarayabilecek temel kavramlar: Temel çizge yapısı, çizgelerde ağaç yapısı, ağaçta dolaşma, derinlik öncelikli arama (**DFS**), genişlik öncelikli arama (**BFS**).

Soruda bize verilenden yola çıkarsak, elimize geçecek olan Marsupilami mahallesinin **yönsüz** ve **ağırlıksız** bir ağaç (**undirected, unweighted tree**) olduğunu rahatlıkla söyleyebiliriz. Bu bilginin ışığında, soru aslında N adet düğümden oluşan bir ağaçta 1 numaralı ve N numaralı düğümler arasındaki mesafeyi bulmamızı istemektedir.

Ağacın yönsüz ve ağırlıksız olduğunu düşünürsek, sorunun çözümü için çizgelerde (graph) yaygın kullanılan arama yöntemlerini kullanabiliriz. Bu açıdan, 1 numaralı düğümden başlayan bir **derinlik öncelikli arama** (Depth First Search - DFS) veya **genişlik öncelikli arama** (Breadth First Search - BFS) algoritması vasıtasıyla N numaralı düğüme ulaşmak ve her adımda 1 numaralı düğümden ne kadar uzaklaştığımızı tutmak çözüm için yeterli olacaktır.

Örnek bir çözüm için aşağıdaki C++ kodunu inceleyebilirsiniz.

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int n;
6  vector<int> edges[100001];
7
8  void dfs(int node, int ata, int level){
9      if(node == n || level > 3) {
10         if(level <= 3) {
11             cout << level;
12         } else {
13             cout << "COK";
14         }
15         exit(0);
16     }
17     // Aşağıdaki for döngüsündeki next degiskeni
18     // for(int i = 0; i < edges[node].size(); i++){
19     //     next = edges[node][i];
20     // }
21     // ifadesi ile aynidir.
22     for(int next : edges[node]){
23         if(next != ata) {
24             dfs(next, node, level + 1);
25         }
26     }
27 }
28
29 int main() {
30     cin >> n;
31
32     for(int i = 1; n > i; i++){
33         int x, y;
34         cin >> x >> y;
35         edges[x].push_back(y);
36         edges[y].push_back(x);
37     }
38
39     dfs(1, 0, 0);
40 }
```