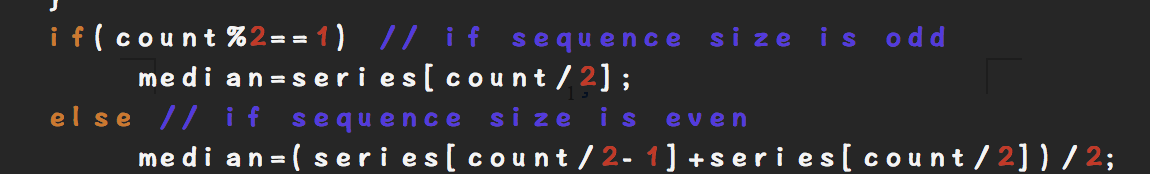
**★★★☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：10938: What is the Median?**

**解題者：周永霖**

**解題日期：2017年6月 15日**

**題意：**

有兩隻跳蚤，他們透過樹枝，在所有樹葉中跳來跳去。題目要找出他們會在哪一個樹葉遇到，若剛好錯開，就會不斷的互跳。  
第一部分要求輸入N個樹葉（樹枝則是N-1枝），接著輸入N-1組連接樹枝的樹葉 ;   
第二部分要求輸入L組跳蚤，接著輸入每一組跳蚤各自的起始位置。

**題意範例：**

input output

8 The fleas meet at 2.

1 2 The fleas meet at 1.

1 3 The fleas jump forever between 2 and 5.

2 4 The fleas meet at 1.

2 5 The fleas jump forever between 1 and 2.

3 6

3 7

5 8

5

5 1

7 4

1 8

4 7

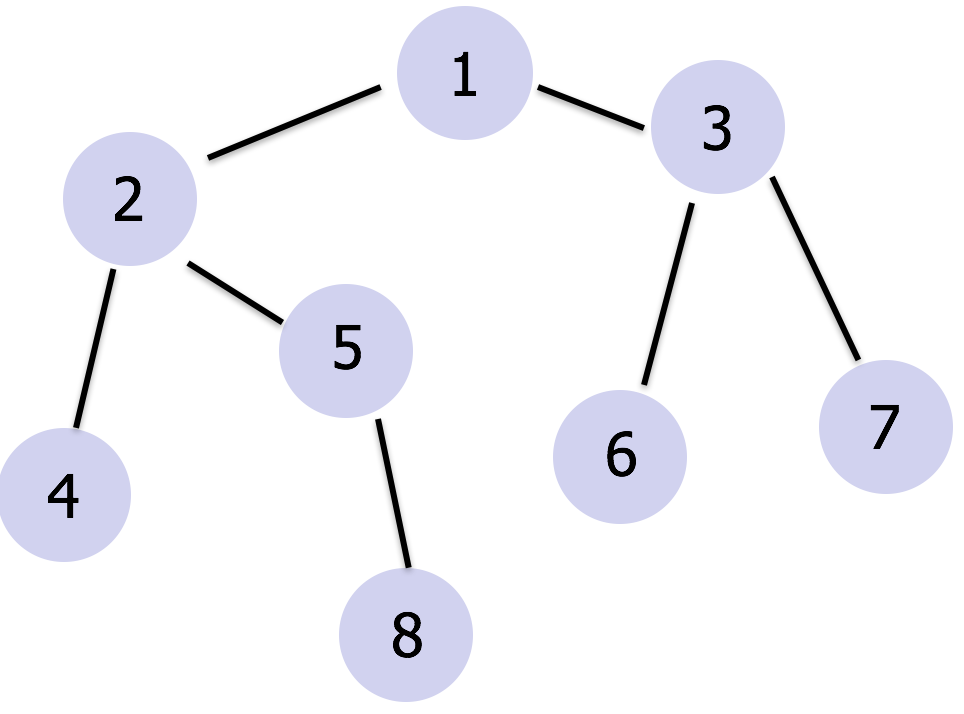
7 8

0

**解法：**

* 先固定一隻flea1，用第二隻flea2遍歷整棵樹  
  （DFS），直到找到固定的flea1。
* 找到後判斷最短路徑長是奇數還偶數，  
  奇數：交錯而過。  
  偶數：剛好會某個node遇到。

**解法範例：**



**討論：**

時間複雜度：DFS，所以O(n2)

**程式：**

#include<iostream>

#include<cstring>

#include<cstdio>

#include<vector>

using namespace std;

vector<int> G[5005]; // store the graph branch

int flea1,flea2; // two fleas first position

bool is\_tra[5005]; // check if is traveled

int path[5005]; // record which node we have traveled

void DFS(int len,int node,bool alrdy\_fd){

    if(is\_tra[node] || alrdy\_fd)

        return ;

    path[len]=node; // store node we travel now

    if(node == flea2){ // if find another flea

        if(len %2 == 0){ // length of path is even

            cout << "The fleas meet at " << path[len/2] << "." <<endl;

        }

        else{ // odd

            if(path[len/2+1] > path[len/2])

                cout << "The fleas jump forever between " << path[len/2] << " and " << path[len/2+1] << "." <<endl;

            else

                cout << "The fleas jump forever between " << path[len/2+1] << " and " << path[len/2] << "." <<endl;

        }

        alrdy\_fd=true;

        return ;

    }

    is\_tra[node]=1; //set this node is traveled

    for(int i=0 ; i<G[node].size() ; i++){

        DFS(len+1,G[node][i],alrdy\_fd); // recursive

    }

}

int main(){

    int N,L;

    while(cin >> N){

        if(N == 0)

            break;

        int x,y;

        for(int i=0 ; i<N-1 ; i++){

            cin >> x >> y;

            G[x].push\_back(y);

            G[y].push\_back(x);

        }

        cin >> L;

        for(int i=0 ; i<L ; i++){

            memset(is\_tra,false,5005);

            memset(path,0,5005);

            cin >> flea1 >> flea2;

            if(flea1 == flea2){

                cout << "The fleas meet at " << flea1 << "." <<endl;

                continue;

            }

            DFS(0,flea1,false);

        }

        for(int i=1 ; i<=N ; i++) // clear the Graph vector in the end

            G[i].clear();

    }

    return 0;

}