Algorithms

BFS Homework 5

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Problem #1: LeetCode 934 - Shortest Bridge

In a given 2D binary array `grid`, there are two islands. (An island is a 4-directionally connected group of `1`s not connected to any other 1s.)

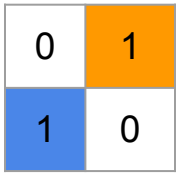
Now, we may change `0`s to `1`s so as to connect the two islands together to form 1 island.

Return the smallest number of `0`s that must be flipped. (It is guaranteed that the answer is at least 1.)

- C++: int shortestBridge(vector<vector<int>> &matrix)
- Java: public int shortestBridge(int[][] grid)
- Python: def shortestBridge(self, grid: List[List[int]]) -> int:
- Javascript: var shortestBridge = function(grid)
- Develop an O(rows x cols) solution

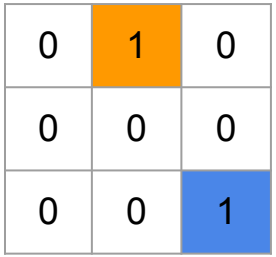
Example 1:

```
Input: grid = [[0,1],[1,0]]
Output: 1
```



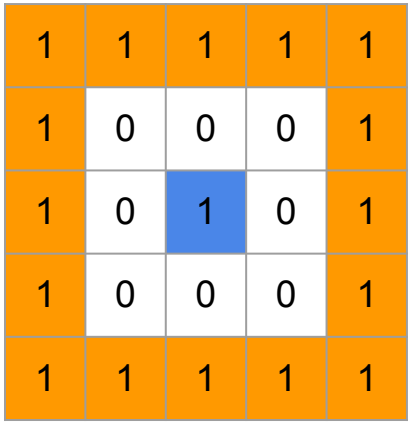
Example 2:

```
Input: grid = [[0,1,0],[0,0,0],[0,0,1]]
Output: 2
```



Example 3:

```
Input: grid = [[1,1,1,1,1],[1,0,0,0,1],
[1,0,1,0,1],[1,0,0,0,1],[1,1,1,1,1]]
Output: 1
```



Problem #2: SPOJ PT07Z - Longest path in a tree

You are given an unweighted, undirected tree. Write a program to output the length of the longest path (from one node to another) in that tree. The length of a path in this case is number of edges we traverse from source to destination.

Input

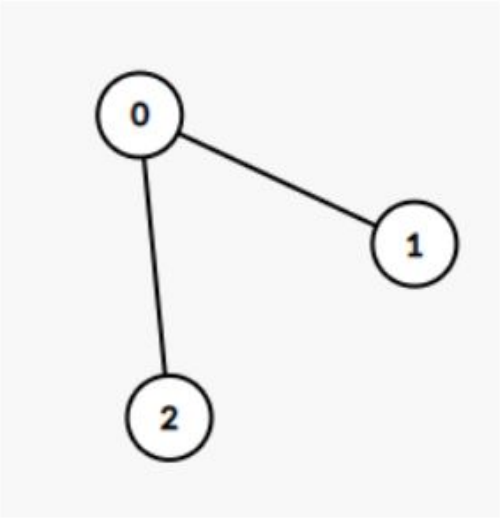
The first line of the input file contains one integer *N* --- number of nodes in the tree (0 < *N* <= 10000). Next *N*-1 lines contain *N*-1 edges of that tree --- Each line contains a pair (*u*, *v*) means there is an edge between node *u* and node *v* (1 <= *u*, *v* <= *N*).

- C++: int treeDiam(GRAPH &adjList)
- Python: def treeDiam(adjList) -> int
- The same as premium problem: LeetCode 1245 - Tree Diameter
- You can use my template: 02_SPOJ_PT07Z_template.cpp

Input:
3
1 2
2 3

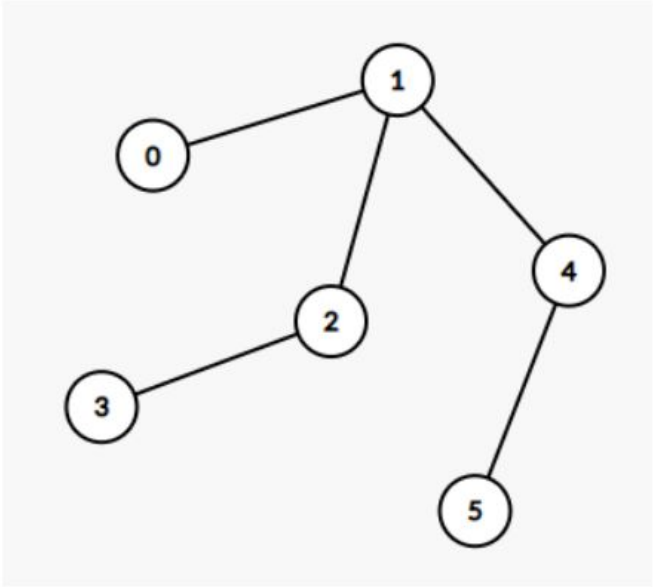
Output:
2

Example 1:




```
Input: edges = [[0,1],[0,2]]
Output: 2
Explanation:
A longest path of the tree is the path 1 - 0 - 2.
```

Example 2:




```
Input: edges = [[0,1],[1,2],[2,3],[1,4],[4,5]]
Output: 4
Explanation:
A longest path of the tree is the path 3 - 2 - 1 - 4 - 5.
```

Note: these examples are from LeetCode website (0-based nodes). Spoj input is 1-based

Requirements

- Find 2 different solutions
  - Different, not just rewriting the same idea with BFS/DFS
- BFS-Based solution
  - Develop a solution based on BFS
  - You can develop a slow version
  - Or think about an optimized version that uses 2 BFS calls
- DFS-based solution
  - We can develop a **single DFS call** to compute the diameter!
  - Hint: start thinking using a **rooted binary** tree.
    - Where is the diameter path relative to the current node during the DFS? 2 simple cases.

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."