Coding Assignment 2

ECS 122A Algorithm Design and Analysis

Diameter of a Tree

Problem Description

You are given a tree T with n vertices numbered $1, 2, \ldots n$.

Find the diameter of the tree. The diameter of a tree is the maximum distance among all pairs of vertices in the tree. The distance between two vertices is the number of edges on the shortest path between them.

(*Hint*). Let u be *any* vertex in the tree and v be the farthest vertex from u. If there are several candidates for v, choose any. Let w be the farthest vertex from v (if there are several such vertices, choose any). The diameter is equal to the distance between v and w. Note that this fact can be used to design a linear time algorithm to compute the diameter.

Remark 1: For this question, your program will be tested against multiple test instances within a single test file. Each test instance contains the description of a tree T.

Remark 2: For this problem, we recommend implementing breadth-first search for finding v and w. If you implement depth-first search using recursion, your program may crash if the recursion is too deep. This is because the maximum call stack size is 10 MB on Gradescope, and the default maximum depth of recursive calls in Python is 1000.

Input and Output

In this assignment, you are provided with sample submissions that contain code for input and output. You only need to complete a function that, given the edges of the tree T (represented by an array of pairs of integers), returns the diameter of T. Your function should ideally be able to compute the diameter in linear time. For completeness, the input/output format is given below.

Each input file contains several test instances. The first line contains a single integer C, representing the number of test instances your program is tested against. The description of the P instances follows.

Each test instance is formatted as follows. The first line contains an integer n, representing the number of vertices in the tree T. Each of the subsequent $n-1$ lines contains a pair of space-separated integers u and v, indicating that there is an edge between vertex u and vertex v in T.

For each test instance, your program should output a single integer: the diameter of the tree T.

Constraints

- Each input file contains multiple test instances.
- $1 \leq C \leq 1000$.
- Each test instance contains a tree of at least 2 vertices.
- In a single input file, the sum of the sizes of all trees is at most $2 \cdot 10^5$.
- The time limit (for all test cases in a single file) is 1 second for C/C++ and 3 seconds for Python.

Test Cases

Your program will be evaluated on 10 visible test files. Passing all the instances within a file will earn you 0.8 points. Note that if your program outputs an incorrect answer for even one of the instances within a file, you will receive 0 points for that file.

Sample Input 1:

```
4
8
1 2
3 1
4 3
1 6
7 5
5 1
4 8
5
3 1
1 2
5 4
4 2
4
2 1
2 3
2 4
2
1 2
```

Sample Output 1:

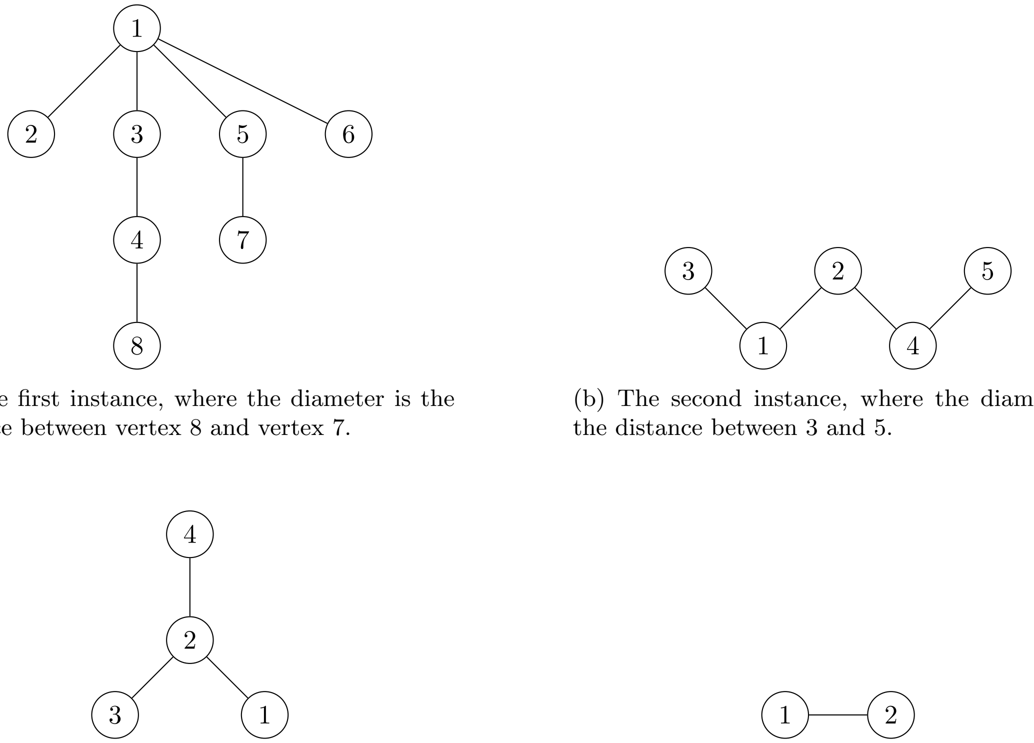
```
5
4
2
1
```

Sample Explanation

Sample Input 1 contains $C = 4$ test instances, which are illustrated in Fig. 1.

Submission Guideline

Write your program in either C, C++ or Python **in a single file**. Submit the file on Gradescope. The time limit on Gradescope is 1 second for C/C++ and 3 seconds for Python. You can make at most 10 submission attempts. You may refer to `sample.cpp` or `sample.py` for sample code that takes input and writes output.



(a) The first instance, where the diameter is the distance between vertex 8 and vertex 7.

(b) The second instance, where the diameter is the distance between 3 and 5.

(c) The third instance, where the diameter is the distance between 1 and 3.

(d) The fourth instance, where the diameter is the distance between 1 and 2.

Figure 1: Illustrations for Sample Input 1