

CS 330 – Fall 2021
Homework 5, Problem 2: Smallest Hole
Due October 6, 2021

Given an **undirected** graph, a *hole* is a cycle with no repeated vertices, and no edges “across” the cycle (that is, you cannot form a shorter cycle using the same nodes).

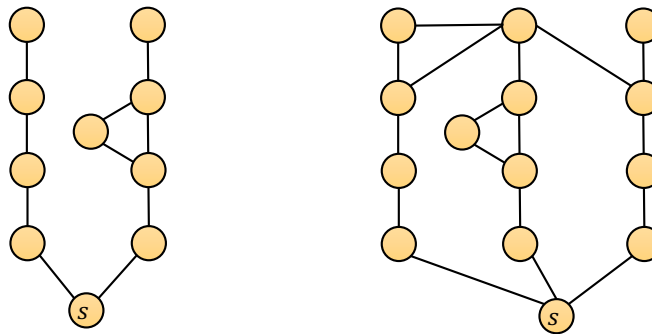
Formally, a hole is a sequence of nodes u_1, u_2, \dots, u_k , all of which are different, that form a cycle (so (u_1, u_k) and all the edges u_i, u_{i+1} , for $i = 1, \dots, k - 1$, are present) but such that no other edges (u_i, u_j) exist (that is, if $i < j - 1$ then (u_i, u_j) is not an edge).

Your task in the assignment is to find the smallest (that is, shortest) hole containing a given vertex, if any hole exists.

Inputs An undirected graph $G = (V, E)$ and source vertex s .

Outputs A Boolean (true/false) indicating whether a hole containing s exists. If a hole exists, you should also find the length of the shortest hole and the list of vertices in such a hole.

For example, in the left-hand graph below, s is not part of any hole. In the right-hand graph, there are three holes containing s of length 8 (your algorithm just has to find one of them).



Design an algorithm for this task and program it in Python. For full credit, your algorithm should run in time $O(m + n)$ where $n = |V|$ and $m = |E|$.

Starter Code We’ve provided starter code that consists of

- **simplegraphs.py** : A collection of functions for manipulating graphs as python dictionaries. There are procedures for reading a graph from a file that describes it, writing the graph to a file, and other basic manipulations. It includes code for BFS and DFS. You can use any of these in your solution. Do not change the file **simplegraphs.py** (since the autograder will include the version we distributed). You can create your own version and modify the functions there if you like.
- **shortestHole.py** : A template for your actual solution. There are functions for reading the input and writing the output (see below). All you have to do is complete the procedure **shortestHole**.
- Test inputs and outputs. You should do your own testing!

Program Format You should submit a file called `shortestHole.py` containing your solution (we'll give you a template to start from). We will test your program by running the following command:

```
python shortestHole.py graph_file source_file output_file
```

There are three arguments, each of which is the name of a file

- `graph_file` describes the graph (see below for the exact format). This uses the format that is read by `simplegraphs.py`. The first two lines of the file list the number of nodes n and the sum of degrees in the graph. The remaining lines list the edges in the graph, assuming the node IDs are $\{1, \dots, n\}$.
- `source_file` lists the source vertex. This file just has a number in it (one node ID).
- `output_file` is where your program will write its output. Code for writing the output is provided. The first line is a Boolean (true/false), the second line is an integer (the shortest hole length, or -1 if no hole was found) and the third line is a list of the IDs of the nodes in the shortest hole.