CS 211 Data Structures and Algorithms Lab
July -- December, 2019
Examination
18 November, 2.30 pm to 5.30 pm
Maximum Marks: 25

Instructions

- You can refer notebooks for this examination. Please do not bring any textbook.
- You will have access to an online copy of the textbook.
- Personal electronic devices are not allowed and the Internet should not be accessed.
- Unlike assignments, please do not have discussion with others.
- Unlike assignments, apart from an output file, you need to manually write a file named "description.txt" which explains the ideas you used for the program. This carries 4 marks.
- Please read this document carefully.

The objective of this exam is to write a program to compute some parameters of an input graph.

Input

Your program should accept one command-line argument which is a file containing the description of a connected, undirected, and unweighted graph. A typical execution of your program will be './a.out input.graph'.

Format of an input file

Input file represents a connected, undirected, and unweighted graph. Every line in the input file will be of the form 'u v' which represents an undirected edge between two distinct vertices with labels 'u' and 'v'. No edges are repeated in the file.

Tasks

You have to write a program which computes a few parameters of the input graph represented by the input file. Following are the parameters to be identified.

1. Number of nodes in the input graph
2. Number of edges in the input graph
3. Minimum degree of the graph
4. Maximum degree of the graph
5. Radius of the graph
6. Diameter of the graph
7. Girth of the graph

Some definitions required to understand the parameters are given below.

- The degree of a vertex is the number of edges incident to that vertex.
- The minimum degree of a graph is the minimum of the degrees of its vertices.
- The maximum degree of a graph is the maximum of the degrees of its vertices.
- The distance between two vertices is the length of a shortest path between them.

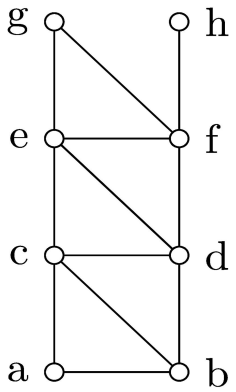
- The eccentricity of a vertex v is the greatest distance between v and any other vertex.
- The radius of a graph is the minimum of eccentricities of the vertices of the graph.
- The diameter of a graph is the maximum of eccentricities of the vertices of the graph.
- A cycle in a graph is a subgraph formed by a sequence of vertices $v_1,v_2,v_3,...,v_k$ with the edges v_1v_2, v_2v_3,...,$v_{k-1}v_k$, v_kv_1.
- The length of a cycle is the number of edges present in the cycle (which is the same as the number of vertices present in the cycle).
- The girth of a graph is the length of a shortest cycle. Note that girth is at least 3. If there is no cycle in the graph, then girth is infinity.

Output

Your program should write a file with name 'parameters.txt'. **The file should have exactly seven lines each line containing exactly one integer.** The required content of each line is described below. The last column shows the marks you secure if your program correctly compute each parameter.

Line no.	**Content of the line**	**Marks**
1	<number of vertices>	1
2	<number of edges>	1
3	<minimum degree>	2
4	<maximum degree>	2
5	<radius>	5
6	<diameter>	5
7	<girth>/-1 if girth is infinity	5

Consider the graph shown below.



An output file for the given graph is described below.

Line no.	Content of the line	Comment
1	8	
2	12	
3	1	The vertex with minimum degree is ‘h’
4	4	The vertices with maximum degree are ‘c’, ‘d’, ‘e’, and ‘f’.
5	2	The minimum of eccentricities is achieved at vertices ‘d’ and ‘e’.
6	4	The maximum of eccentricities is achieved at vertices ‘a’ and ‘h’.
7	3	Shortest cycles in this graph are triangles.

Additionally, you have to manually create a file named “description.txt” which explains the ideas and algorithms you used for the program. How each of the tasks is accomplished is to be described in the file. Writing pseudocode is not mandatory but conveying ideas is mandatory. You don’t have to be too elaborate. You may update this file after completing each task (instead of updating everything at the end). **This carries 4 marks.**

Sample inputs and outputs

You are given with some sample inputs and outputs. The names of the input file for a particular test is input**x**.graph. Test your program as ./a.out inputx.graph and compare the output of it with the given output file parameters**x**.txt.

Submission and Evaluation

- The program you submit should output a file named parameters.txt’ when run. Additionally, you have to manually create a file named “description.txt”.
- During evaluation, we will terminate your program if your program takes more than 5 minutes in a typical lab machine. The size of the input graph that will be used for evaluation will be similar to some of the sample input graphs.
- There should be only one main file and it should be named as <roll no>.<extension>, where roll no. specifies your roll no. and the extension depends on the language you choose (Usage of C/C++ is mandatory for this exam). Ex: 180040001.c
- Test well before submission. You may use the attached sample input files for testing. The corresponding output files are also attached. We have some hidden inputs with us to test your program. The mark you obtain is purely based on whether your program correctly gives outputs for the hidden inputs.
- If your program has only a single source file, please submit the file as it is. If your program has multiple source files, please submit your code as a zip file where the name

of the zip file should be your roll number. You can submit the description.txt file separately. It is important that you follow the input/output conventions exactly (including the naming scheme) as we may be doing an automated evaluation. There will be a penalty of 10% (on the mark you deserve otherwise) if you do not follow the naming conventions exactly.

- Follow some coding style uniformly. Provide proper comments in your code.
- Submit only through moodle. Submit well in advance. Any hiccups in the moodle/internet at the last minute is never acceptable as an excuse for late submission. Submissions through email or any other means will be ignored.
- Copying others' programs and allowing others to copy your program are serious offences and deserving penalty will be imposed if found.