

CS 211 Data Structures and Algorithms Lab
Autumn 2022

Assignment no.	8
Objective	To implement BFS and to use it to solve single-source shortest path problem on undirected graphs
Total marks	10
Due date (without penalty)	17th October (Monday) 11:59 pm
Cut-off date (with penalty - 5%)	20th October (Thursday) 11:59 pm
Penalty for violating naming convention(s)	5%

Command-line argument:

Your program should receive a file (input file) as a command line argument.

Input:

Your program should accept an input file as a command-line argument. A typical execution of your program will be ***./a.out input.graph***

- The input file represents an undirected graph.
- The first line of the file contains two numbers:
 - (i) the number of vertices n ;
 - (ii) the number of edges m .
- The vertices are numbered from 0 to $n-1$. Every other line is of the form: $x \ y$, which represents an undirected edge between *vertex x and vertex y* .
- No edge is repeated in the input file.
- The vertex 0 is the source (starting) vertex from which you need to find the shortest distances to other vertices.

Your program may create an adjacency list of the input graph.

Task:

Implement BFS and use it to find the shortest distance of every vertex from the source vertex (0). The output file should be named as '***sd.txt***'. Every line in the output file should have exactly one integer which represents the distance between the source vertex and the vertex with label $k-1$, where k is the line number. For example, *the first line should have the distance between the source vertex 0 and the vertex 0 - of course, this value is always 0* . The output file must have exactly n lines. The graph that will be used for evaluation will have at least 1000 vertices and at most 2000 vertices.

Note#1: if there is no path exists from the source vertex 0 to any other given vertex (means if no shortest path exists from the source vertex), then the output file should contain “-1” at that vertex

Note#2: if there is no vertex is presented in the given input file, in that case also the output file should contain value “-1” at that vertex

Submission

- The program you submit should output '**sd.txt**' when run.
- The main file of your program 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 is mandatory for this assignment). Ex: 210010001.c.
- Do the stress test of your program well before submission.
 - (i) You may use the attached sample input files for testing; the corresponding output files are also attached;
 - (ii) 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. 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 5% (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 at the last minute is never acceptable as an excuse for late submission. Submissions through email or any other means will be ignored.
- Acknowledge the people (other than the instructor and TA) who helped you to solve this assignment. The details of the help you received and the names of the people who helped you (including internet sources, if applicable) should come in the beginning of the main file as a comment. **Copying others' programs is a serious offense and a deserving penalty will be imposed if found.**

Evaluation

- To consider for the first evaluation without penalty, you have to submit your program by the due date. If you submit after the **due date** but on or before the **cut-off date**, **there will be a penalty of 5% on the marks you deserve otherwise.**
- If you do not submit by the **cut-off date**, your program will not be considered for the first evaluation.
- We will do the first evaluation after the cut-off date. The marks you obtain will be proportional to the number of correct lines in the output files. We will not be using 'diff' for the evaluation of this assignment. Instead we will count the number of correct lines in the

output file and your marks will be proportional to that.

- We will do the second evaluation later. This is for those who want to improve their marks obtained in the first evaluation or who do not submit for the first evaluation. There will be **a penalty of 20%** for those who are submitting for the second evaluation. The details of the second evaluation will be shared later.