National Taiwan University
Graduate Institute of Electronic Engineering
VLSI Physical Design, Fall 2020

Guidelines for Programming Assignment #1
(due on-line at 5pm, Oct 14, 2020)

***Submission*: email to Prof. Chen, csj@ntu.edu.tw**

Problem

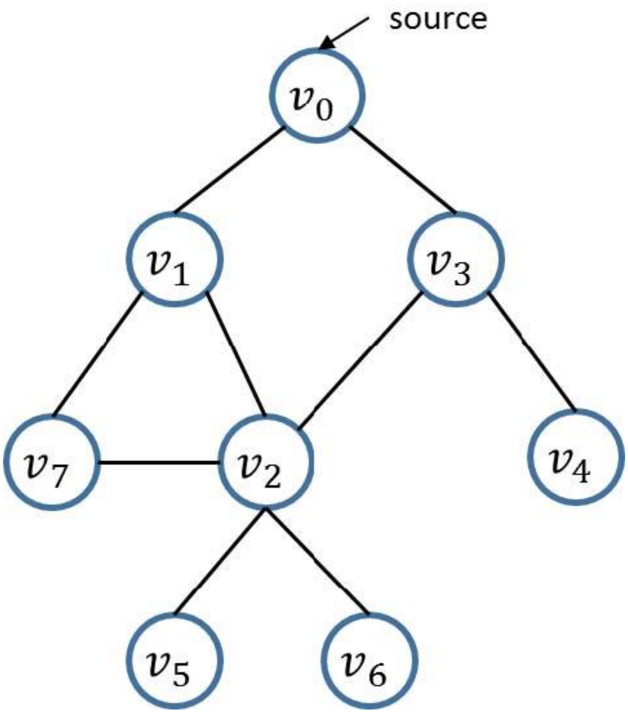
You are asked to apply the **Breadth-first search (BFS)** to traverse every node and print the BFS tree of an undirected connected graph.

Input

The first line of the input file is the number of vertices of the test-case. The second line is the source vertex where your BFS algorithm should start from. The pairs of vertices represent edges. For example, “.edge 2 5” represents an edge between vertex 2 to vertex 5. Below is an example of the input format.

```
// This is comment
//Input:
.number_of_vertex 8
.source 0

.edge 0 1
.edge 0 3
.edge 3 4
.edge 3 2
.edge 1 2
.edge 1 7
.edge 2 7
.edge 2 6
.edge 2 5
```

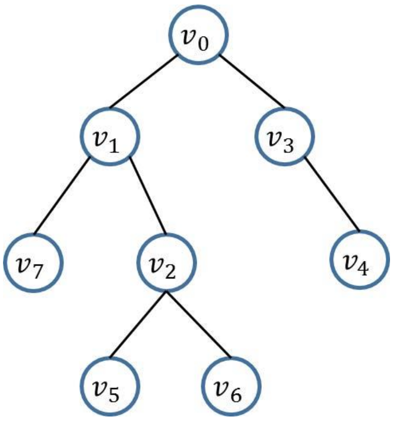


Output

1. **Please print the BFS tree in the order of traversal**. When performing BFS, if there are multiple vertices to choose, your algorithm must choose the **vertex with minimal index** first. Below is an example of BFS, the traversal starts from v_0, and then v_1 should be visited earlier than v_3. Therefore, edge (v_0,v_1) should be printed before edge (v_0,v_3). When printing the edges, your program should print the predecessor vertex in front of the successor vertex. The following graph is an example, where v_0 is visited before v_1, so your program should print ".edge 0 1" instead of ".edge 1 0".

The output file *MUST* be in the required format so that they can be verified automatically. Failing to follow the format might incur zero scores or significant penalties.

```
// Output:
.edge 0 1
.edge 0 3
.edge 1 2
.edge 1 7
.edge 3 4
.edge 2 5
.edge 2 6
```



2. Please refer to the files in output/ directory: v8.out

Required Files

You need to submit the following materials in a .tar.gz or .zip file:

(1) Source codes and Makefile

(2) Executable binary

(3) A text readme file describing how to compile and run your programs

(4) A report (**report.doc**) includes:

1. The CPU time and memory usage used by the **BFS** algorithm using the following table:

2. The data structure you used in your program.

Test Case	BFS	
	Time (Sec)	Memory (MB)
v8		
v100		
v1000		
v10000		
v30000		

Language/Platform

Language: C or C++. It is better that your program can be executable on the EDA union Lab machine.

Please visit the following website for the information on the EDA Union account application: http://edaunion.ee.ntu.edu.tw/.

File-name Rule

The submission filename should be <student_id>-<p1>.tar.gz or <student_id>-<p1>.zip (e.g. **r02943000-p1.zip**). If you have a modified version, please add _v[version_number] as a postfix to the filename and resubmit it to the submission website (e.g. . **r02943000-p1-v1.zip**, **r02943000-p1-v2.zip**, etc.).

Command-line Parameter

You have to add command-line parameters in your program to specify the input and output file name as the format (e.g., `BFS input.dat output.dat`):

```
[executable_file_name] [input_file_name] [output_file_name]
```

Evaluation

The individual score per test case is determined by the

(1) correctness

(2) output file format

(3) report.doc

(4) demo in class

  -- Run hidden cases

If there is any question, please email to csj@ntu.edu.tw

.