

INSTRUCTIONS - PRACTICAL EXAM – CSD201

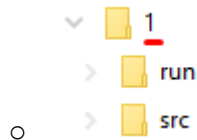
Read the instructions below carefully before start coding.

Students are **ONLY** allowed to use:

- Software tools must be used: **NetBeans IDE 8.x** and **Java JDK 1.8**.
- His / her own study materials like presentation slides, notes, sample codes, program examples, electronic books stored on his / her computer only.
- For distance learning: Google Meet, Hangout (for Exam Monitoring Purpose).

Instructions

- Step 1: run “Clean and Build Project” (Shift+F11) to create **dist** folder and **.jar** file.
- Step 2: Prepare to submit answer:
 - For each question (e.g., question **1**), please create two sub-folders: **run** and **src**.
 - Copy *.**jar** file into **run** folder
 - Compress source code into **.zip**, then copy **.zip** file into **src** folder.
- Step 2: Submit solution for each question:
 - Choose question number (e.g., **1**) in PEA software, and then attach corresponding solution folder (e.g., **1**). Click Submit button to finish submitting this question.



Notes

- **Do not use accented Vietnamese** when writing comments in programs.
 - Solutions will be marked by Automated Marking Software.
 - **If at least one of the above requirements is not followed, the exam will get ZERO.**
-

Question 1: (3 marks)

Write a program to build an **undirected** graph by giving edge list.

Giving a start vertex, your task is to DFS (Depth First Search) and BFS (Breadth First Search) the given undirected graph.

Example:

- Giving the edge list that used to represent undirected graph as follow:

```
8 10 0
0 1
0 2
0 5
1 6
2 3
2 4
3 4
4 5
4 6
5 7
```

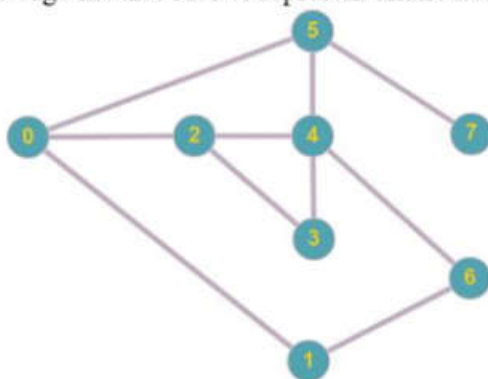


Figure 1. The undirected graph that created by

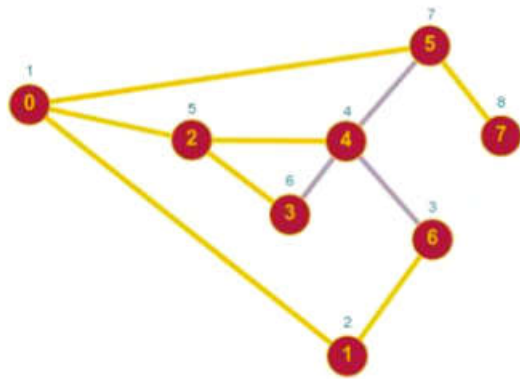


Figure 2. The DFS traversing of the graph is:
0, 1, 6, 4, 2, 3, 5, 7

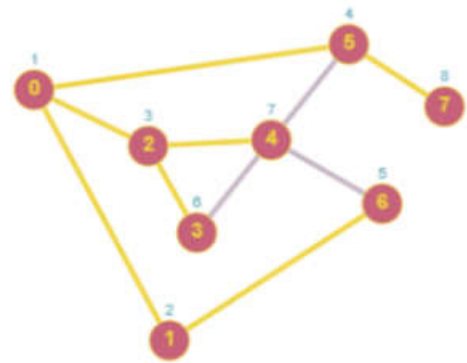


Figure 3. The BFS of the graph is:
0, 1, 2, 5, 6, 3, 4, 7

The input: are stored in the *input.txt* text file:

- The first line contains a 3 integers including:
 - N ($1 \leq N \leq 30$) is the number of vertex.
 - M ($1 \leq M \leq 435$) is the number of edge.
 - S ($0 \leq S \leq N-1$) is the starting vertex.
- The next M line, each line contains 2 integers u and v that represent the edge (u, v) of the graph.

The output: the results need to be saved to the *output.txt* text file:

The first line contains the list of numbers representing the DFS traversing of the graph. Each number separated by one comma.

The second line contains the list of numbers representing the BFS traversing of the graph. Each number separated by one comma.

Sample Input 1	Sample Output 1
8 10 0 0 1 0 2 0 5 1 6 2 3 2 4 3 4 4 5 4 6 5 7	0, 1, 6, 4, 2, 3, 5, 7 0, 1, 2, 5, 6, 3, 4, 7

Sample Input 2	Sample Output 2
6 7 3 0 1 0 2 0 5 2 3 2 4 3 4 4 5	3, 2, 0, 1, 5, 4 3, 2, 4, 0, 5, 1

Question 2: (4 marks)

Write a program to build a binary search tree by inserting N ($1 \leq N \leq 100$) integer values into the BST one by one. Note that the nodes **can have duplicate values**.

Your task is to display the in-order and pre-order traversing of the BST.

For example,

- Create a BST tree by adding into the tree N integer values as follows: 7, 9, 4, 9, 1, 12, 6, 7, 1, 10.

- The in-order traversing of the BST tree is: 1, 1, 4, 6, 7, 7, 9, 9, 10, 12.
- The pre-order traversing of the BST tree is: 7, 7, 4, 1, 1, 6, 9, 9, 12, 10.

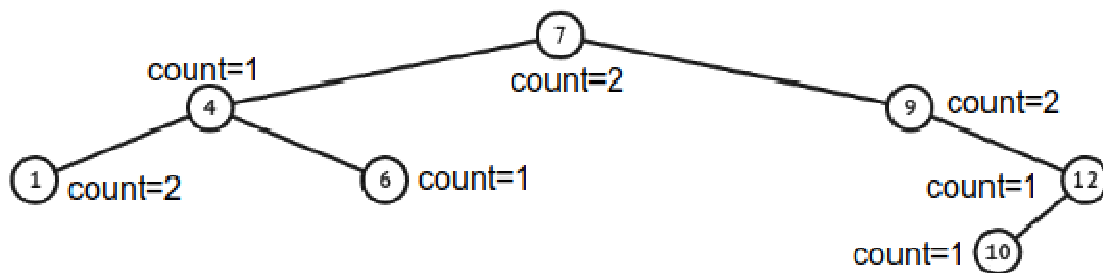


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

The input: are stored in the *input.txt* text file:

- The first line contains a positive integer N ($1 \leq N \leq 100$) which is the number of integer values to insert into the BST.
- The second line containing N integers that will be inserted into the BST one by one, each number separated by at least one space.

The output: the results need to be saved to the *output.txt* text file:

- The first line contains the list of numbers representing the **in-order** traversing of the BST. Each number separated by one comma.
- The second line contains the list of numbers representing the **pre-order** traversing of the BST. Each number separated by one comma.

Sample Input 1	Sample Output 1
10	1, 1, 4, 6, 7, 7, 9, 9, 10, 12
7 9 4 9 1 12 6 7 1 10	7, 7, 4, 1, 1, 6, 9, 9, 12, 10

Sample Input 2	Sample Output 2
10	18, 21, 30, 36, 55, 74, 78, 91, 98, 100
78 21 36 18 30 91 74 55 98 100	78, 21, 18, 36, 30, 74, 55, 91, 98, 100

Question 3: (3 marks)

Write a program to build an **undirected** graph by giving adjacency matrix.

Your task is calculating degree of all vertices.

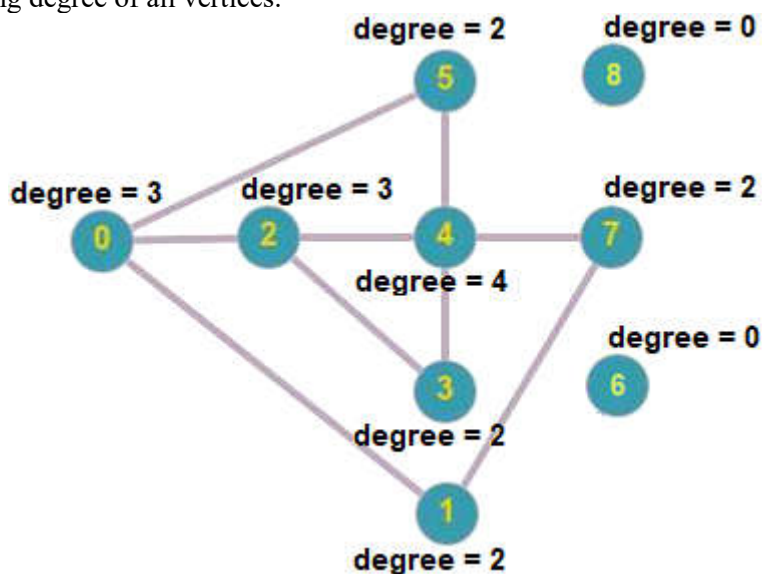


Figure 5. The undirected graph that created by giving adjacency matrix

The degree of all vertices in the short format is: 0(3), 1(2), 2(3), 3(2), 4(4), 5(2), 6(0), 7(2), 8(0).

The input: are stored in the *input.txt* text file:

- The first line contains a 2 integers including:
 - N ($1 \leq N \leq 30$) is the number of vertex.
 - M ($1 \leq M \leq 435$) is the number of edge.
- The next M line, each line contains 2 integers u and v that represent the edge (u, v) of the graph.

The output: the results need to be saved to the *output.txt* text file:

- Only one line contains the list of vertex and its degree, each number separated by one comma.

Sample Input 1	Sample Output 1
9 9 0 1 0 2 0 5 1 7 2 3 2 4 3 4 4 5 4 7	0 (3) , 1 (2) , 2 (3) , 3 (2) , 4 (4) , 5 (2) , 6 (0) , 7 (2) , 8 (0)

Sample Input 2	Sample Output 2
6 7 0 1 0 2 0 5 2 3 2 4 3 4 4 5	0 (3) , 1 (1) , 2 (3) , 3 (2) , 4 (3) , 5 (2)