

Lab 9 - Graph

- Write a program that allows the user to input the edges of a graph and represent the graph as an adjacency matrix.

With std in the format is:

number of vertices

number of edges

Next n lines define the edge: <source node> <target node>

For example, when the user enters:

graph.add_edge(0, 1)

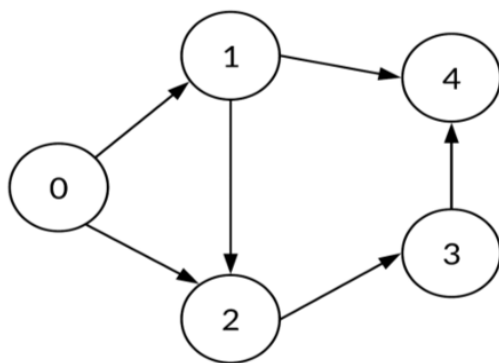
graph.add_edge(0, 2)

graph.add_edge(1, 2)

graph.add_edge(1, 4)

graph.add_edge(2, 3)

graph.add_edge(3, 4)



Adjacency Matrix

	0	1	2	3	4
0	0	1	1	0	0
1	0	0	1	0	1
2	0	0	0	1	0
3	0	0	0	0	1
4	0	0	0	0	0

Std in	Std out
5	0 1 1 0 0
6	0 0 1 0 1
0 1	0 0 0 1 0
0 2	0 0 0 0 1
1 2	0 0 0 0 0
1 4	
2 3	
3 4	

2. Deep first search(DFS). Use DFS to traverse all the nodes in the graph. Print the result to the std out.

In case there are two nodes with the same opening priority and do not know which node to open, the order of priority is 0, 1, 2, 3, ...

Use the graph defined in question 1. Students do not need to enter std in.

Std in	Std out
	0 1 2 3 4

3. Breadth first search (BFS). Use BFS to traverse all the nodes in the graph. Print to the corresponding screen.

In case there are two nodes with the same opening priority and do not know which node to open, the order of priority is 0, 1, 2, 3, ...

Use the graph defined in question 1. Students do not need to enter std in.

Std in	Std out
	0 1 2 4 3

4. Write a program to calculate *in degree* and *out degree*.

In degree: number of edges entering that node.

Out degree: number of edges coming out from that node.

Print to the std out with the content:

<node name> <in degree> <out degree>

Std in	Std out
	0 0 2 1 1 2 2 2 1 3 1 1 4 2 0

5. Write a program to check if there is a path from S to G. Using DFS. algorithm

Print the path or “non-exist path” respectively.

In case there are two nodes with the same opening priority and do not know which node to open, the order of priority is 0, 1, 2, 3, ...

Std in	Std out
0 4	0 1 2 3 4
4 0	non-exist path

6. Write a program to check if there is a path from S to G. Using BFS. algorithm

Print the path or “non-exist path” respectively.

In case there are two nodes with the same opening priority and do not know which node to open, the order of priority is 0, 1, 2, 3,...

Std in	Std out
0 4	0 1 4
4 0	non-exist path