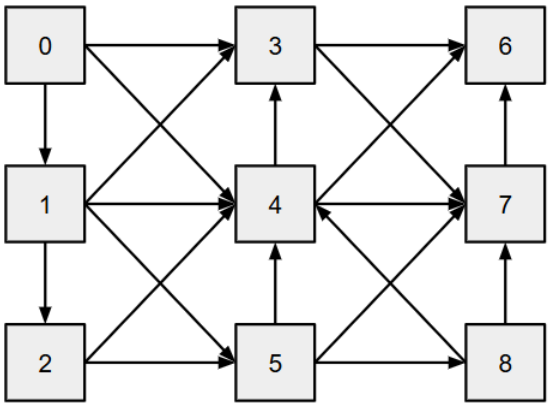
Name: _____

1. Directed Graphs

This problem was adapted from Princeton University's COS 226 course.

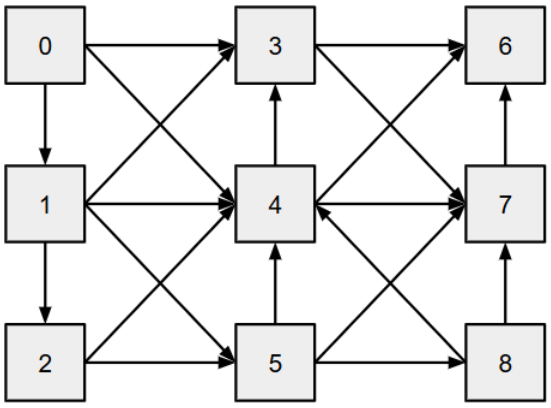
Suppose we had the following graph:



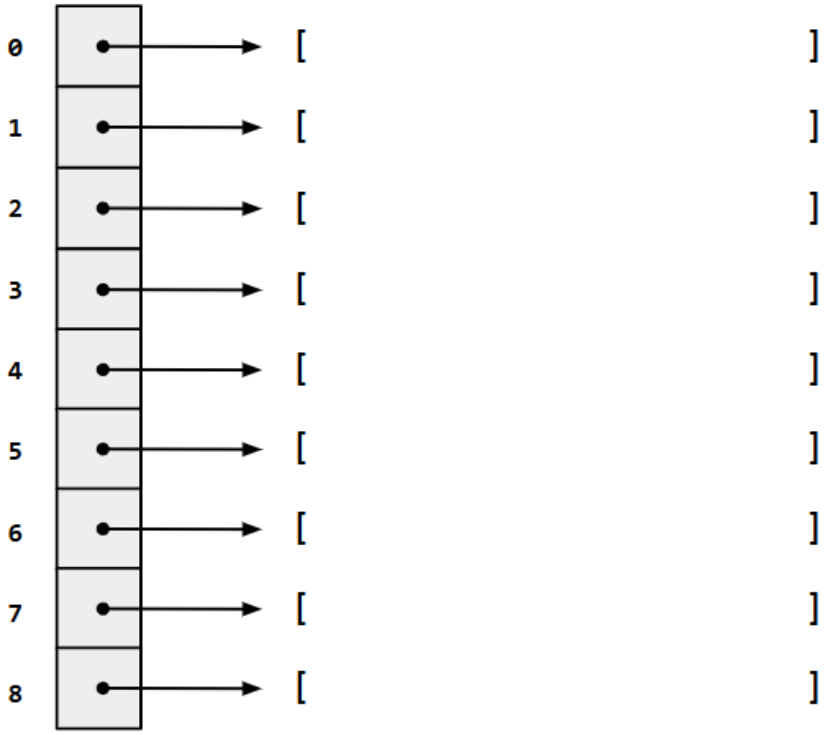
a. Represent the graph above using an adjacency matrix.

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

The following graph is copied over from the first page for your convenience.



b. Represent the graph above using an adjacency list.



c. Perform a depth-first search (DFS) on the graph starting from node `0`. Write down the order in which vertices are visited. Break ties by choosing the smaller integer value.

DFS Pre-Order: _____ _____ _____ _____ _____ _____ _____ _____ _____

d. Perform a breadth-first search (BFS) starting at node `0` and list the order in which nodes are visited. Break ties by choosing the smaller integer value.

BFS Order: _____ _____ _____ _____ _____ _____ _____ _____ _____

2. Depth-First Search Code

Suppose we represent the following graph using an ***adjacency matrix***. The vertices are numbered `0` through `5`, and an entry of `1` indicates that an edge exists between two vertices.

	0	1	2	3	4	5
0	0	1	1	0	0	0
1	1	0	0	1	1	0
2	1	0	0	0	1	0
3	0	1	0	0	0	1
4	0	1	1	0	0	1
5	0	0	0	1	1	0

a. Fill in the blanks below to complete the recursive implementation of depth-first search.

```
def dfs(graph, vertex, marked):

    __________________________ = __________________________

    print(__________________________)

    for neighbor in range(__________________________):

        if __________________________ and __________________________:

            __________________________

marked = [False] * len(graph)
dfs(graph, 0, marked)
```

b. Using your completed implementation from **a**, what is the order in which the vertices are printed?

c. Using the adjacency matrix representation of the graph, perform a breadth-first search (BFS) starting at vertex `0`. List the order in which the vertices are visited. If there is a tie, visit the smaller-numbered vertex first.

BFS Order: _____ _____ _____ _____ _____ _____

3. Breadth-First Search Code

Suppose we represent the following graph using an **adjacency matrix**. The vertices are numbered `0` through `5`, and an entry of `1` indicates that an edge exists between two vertices.

	0	1	2	3	4	5
0	0	1	0	1	0	0
1	1	0	1	0	1	0
2	0	1	0	0	0	1
3	1	0	0	0	1	0
4	0	1	0	1	0	1
5	0	0	1	0	1	0

a. Fill in the blanks below to complete the breadth-first search implementation.

Hint: The `pop` method takes an index as a parameter, removes the element at that index from the list, and returns the removed element.

```
def bfs(graph, start):

    marked = [False] * ________________________

    queue = [________________________]

    ________________________ = ________________________

    while ________________________:

        vertex = ________________________

        print(________________________)

        for neighbor in range(len(graph)):

            if ________________________ and ________________________:

                ________________________

                ________________________

bfs(graph, 0)
```

b. Using your completed implementation from **a**, what is the order in which the vertices are printed?

c. Perform a depth-first search (DFS) starting at vertex `0`. List the order in which the vertices are visited. If there is a tie, visit the smaller-numbered vertex first.

DFS Order: _____ _____ _____ _____ _____ _____