# CS 2420 Algorithms and Data Structures

Fall Semester, 2021

## Assignment 8: Graph Algorithms I

**Due Date:** 11:59 p.m., Tuesday, Nov. 30, 2021

(**Note:** This assignment has four writing exercises and two programming exercises. Please start early because this assignment might be more time-consuming than the previous ones.)

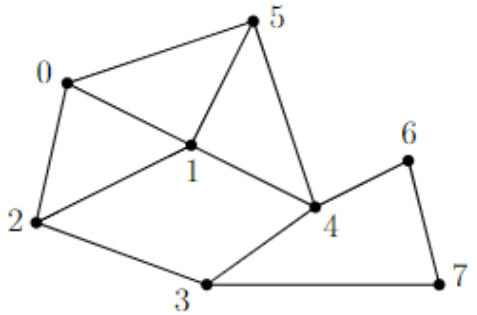



Figure 1: An undirected graph: the numbers besides the nodes are the indices of the nodes.

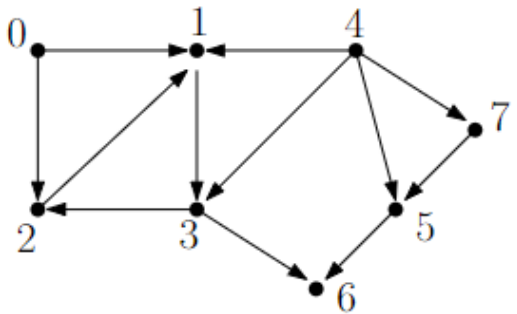



Figure 2: A directed graph: the numbers besides the nodes are the indices of the nodes.

1. Consider the undirected graph in Figure 1. Please give the adjacency lists of the graph. For each list, please order the vertices in the ascending order of their indices. For example, the adjacency list of vertex 0 should be $1 \to 2 \to 5$. **(10 points)**

0 : 1 -> 2 -> 5

1 : 0 -> 2 -> 4 -> 5

2 : 0 -> 1 -> 3

3 : 2 -> 4 -> 7

4 : 1 -> 3 -> 5 -> 6

5 : 0 -> 1 -> 4

6 : 4 -> 7

7 : 3 -> 6

2. Consider the directed graph in Figure 2. Please give the adjacency lists of the graph. For each list, please order the vertices in the ascending order of their indices. **(10 points)**

0: 1 -> 2
1: 3
2: 1
3: 2 -> 6
4: 1 -> 3 -> 5 -> 7
5: 6
6:
7: 5

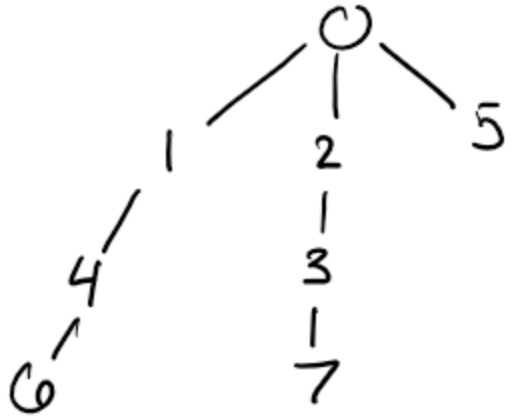
3. Apply the BFS traversal algorithm on the graph in Figure 1 with vertex 0 as the starting vertex and using the adjacency lists in Question 1 (i.e., the vertices in each adjacent list are in ascending order of their indices). **(15 points)**

(a) Please give the BFS traversal order of the vertices that are visited by the algorithm.

0: 1 -> 2 -> 5
1: 4
2: 3
5:
4: 6
3: 7

=> 0 - 1 - 2 - 5 - 4 - 3 - 6 - 7
order of visiting

(b) Please give the BFS tree (which is also the shortest path tree) produced by the algorithm.




BFS tree

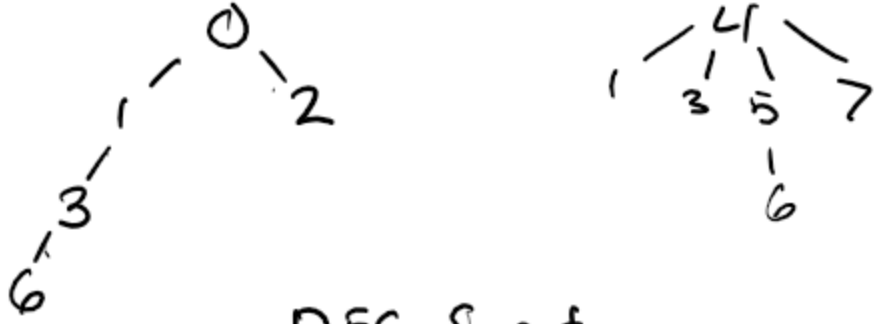
4. Apply the DFS traversal algorithm on the graph in Figure 2 using the adjacency lists in Question 2 (i.e., the vertices in each adjacent list are in ascending order of their indices). Your algorithm should start from vertex 0. Since this is a directed graph, the DFS traversal from vertex 0 may not be able to reach all vertices. Hence, as discussed in class, after the DFS traversal from 0, your algorithm should check other vertices in their index order, and if there is an unvisited vertex, then start a new traversal from that vertex. **(15 points)**

(a) Please give the DFS traversal order of the vertices that are visited by the algorithm.

0: 1 -> 2
1: 3
3: 2 -> 6
2: 1
6:
New entry @ 4
4: 1 -> 3 -> 5 -> 7
5: 6
7:

0 - 1 - 3 - 2 - 6 - 4 - 5 - 7
order of each first visit

(b) Please give the DFS tree/forest generated by the algorithm. Again, if the DFS traversal from vertex 0 cannot reach all vertices, then the algorithm will generate a forest (i.e., multiple trees) rather than a single tree.




DFS forest