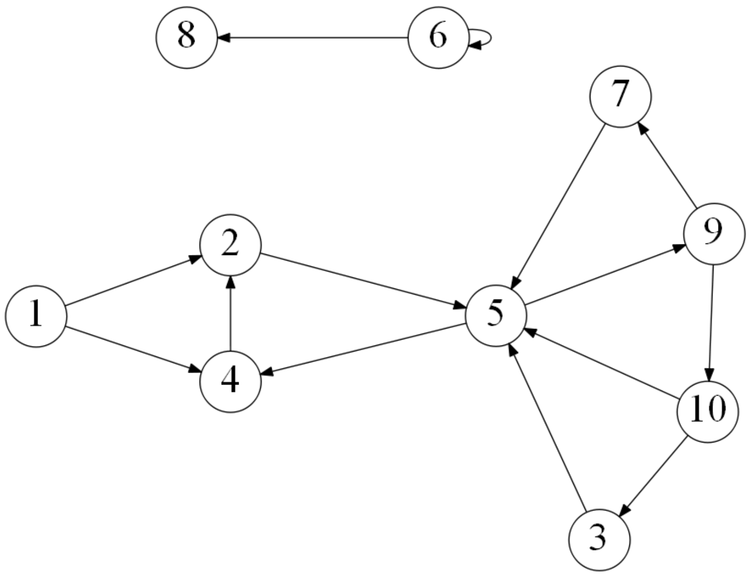
Name: Jessica Noel Date: September 13, 2020

"I pledge my honor that I have abided by the Stevens Honor System."

Point values are assigned for each question. Points earned: ____ / 100

Consider the following graph:



1. Draw how the graph would look if represented by an adjacency matrix. You may assume the indexes are from 1 through 10. Indicate 1 if there is an edge from vertex A -> vertex B, and 0 otherwise. (10 points)

0	1	2	3	4	5	6	7	8	9	10
1	0	**1**	0	**1**	0	0	0	0	0	0
2	0	0	0	0	**1**	0	0	0	0	0
3	0	0	0	0	**1**	0	0	0	0	0
4	0	**1**	0	0	0	0	0	0	0	0
5	0	0	0	**1**	0	0	0	0	**1**	0
6	0	0	0	0	0	**1**	0	**1**	0	0
7	0	0	0	0	**1**	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	**1**	0	0	**1**
10	0	0	**1**	0	**1**	0	0	0	0	0

2. Draw how the graph would look if represented by an adjacency list. You may assume the indexes are from 1 through 10. (10 points)

   $1 \rightarrow 2 \rightarrow 4$
   $2 \rightarrow 5$
   $3 \rightarrow 5$
   $4 \rightarrow 2$
   $5 \rightarrow 4 \rightarrow 9$
   $6 \rightarrow 6 \rightarrow 8$
   $7 \rightarrow 5$
   $8 \rightarrow$
   $9 \rightarrow 7 \rightarrow 10$
   $10 \rightarrow 3 \rightarrow 5$

3. List the order in which the vertices are visited with a breadth-first search. If there are multiple vertices adjacent to a given vertex, visit the adjacent vertex with the lowest value first. (10 points)
   1, 2, 4, 5, 9, 7, 10, 3, 6, 8

4. List the order in which the vertices are visited with a depth-first search. If there are multiple vertices adjacent to a given vertex, visit the adjacent vertex with the lowest value first. (10 points)
   1, 2, 5, 4, 9, 7, 10, 3, 6, 8

5. a) What is the running time of breadth-first search with an adjacency matrix? (5 points)
   θ(V^2)
   b) What is the running time of breadth-first search with an adjacency list? (5 points)
   θ(V+E)

6. a) What is the running time of depth-first search with an adjacency matrix? (5 points)
   θ(V^2)
   b) What is the running time of depth-first search with an adjacency list? (5 points)
   θ(V+E)

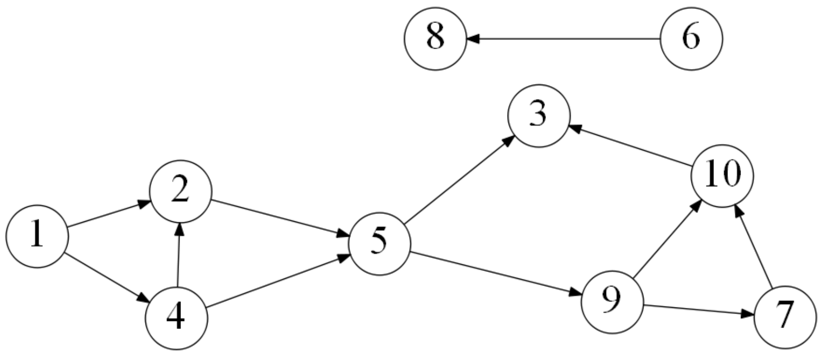
7. While an adjacency matrix is typically easier to code than an adjacency list, it is not always a better solution. Explain when an adjacency list is a clear winner in the efficiency of your algorithm? (5 points)
   An adjacency list is clearly the winner in efficiency when there is a larger data set because the time complexity of the adjacency list (V+E) is faster than that of the adjacency matrix (V^2).

8. Explain how one can use a breadth-first to determine if an undirected graph contains a cycle. (10 points)
   One can use breadth-first to determine if an undirected graph contains a cycle by looking if a vertex has an adjacent vertex that's been checked. If that checked vertex is not the original vertex's parent, then the undirected graph contains a cycle.

9. On undirected graphs, does either of the two traversals, DFS or BFS, always find a cycle faster than the other? If yes, indicate which of them is better and explain why it is the case; if not, draw two graphs supporting your answer and explain the graphs. (10 points)
   In undirected graphs, BFS will find a cycle faster than DFS. DFS will traverse the entire graph to the bottom first, whereas BFS goes through the "top" of the tree first. Therefore, if we were looking to find a cycle using BFS will allow us to stay closer to already checked vertices where its more likely to find a cycle.

10. Explain why a topological sort is not possible on the graph at the very top of this document. (5 points)
    Topological sort is not possible on the graph at the very top of this document because the graph contains a cycle between 2, 4, 5 (there are other cycles within the graph as well). In this case we will never be able to remove these vertices one by one because the indegree is not zero.

Consider the following graph:



11. List the order in which the vertices are visited with a topological sort. Break ties by visiting the vertex with the lowest value first. (10 points)
    1, 4, 2, 5, 6, 8, 9, 7, 10, 3