Name: Himanshu Rana   Date: 10/15/18

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)

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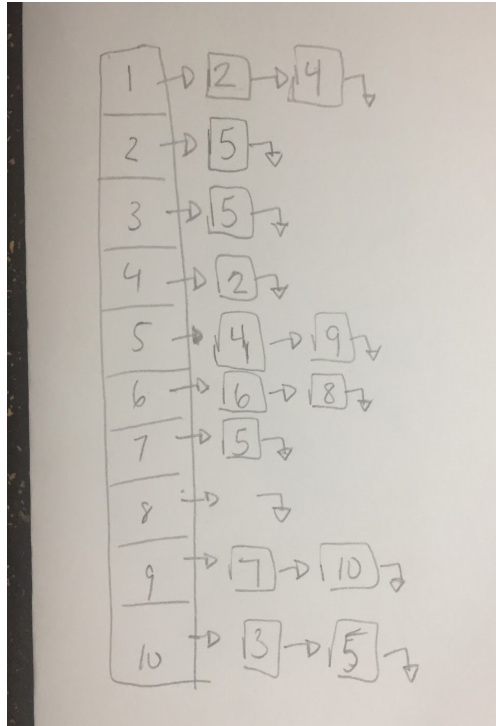
3. List the order in which the vertices are visited with a breadth-first search. If there are multiple vertices adjacent to a given vertex, visit the adjacent vertex with the lowest value first. (10 points)
    1, 2, 4, 5, 9, 7, 10, 3, 6, 8

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 a better algorithm when dealing with large and sparse graphs.

8. Explain why a topological sort is not possible on the graph above. (5 points)
   A topological sort is not possible on the graph above because it contains a cycle.

9. 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