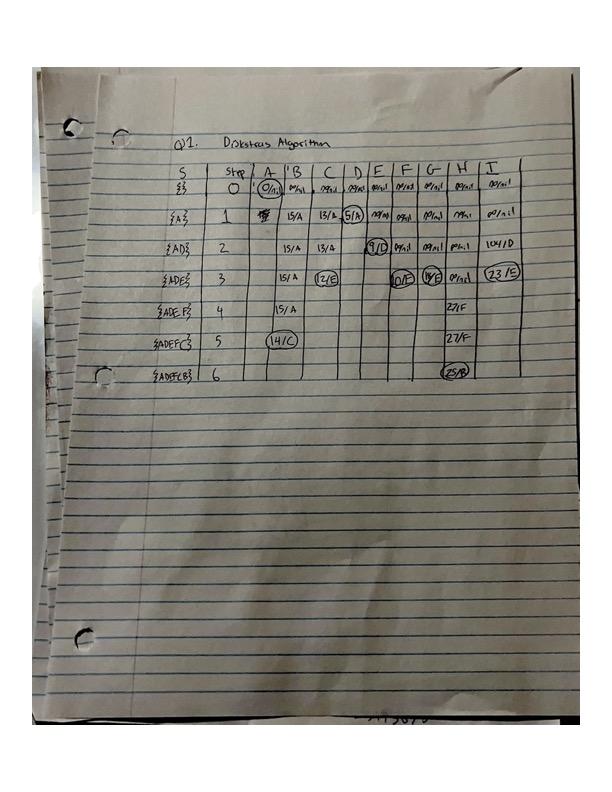
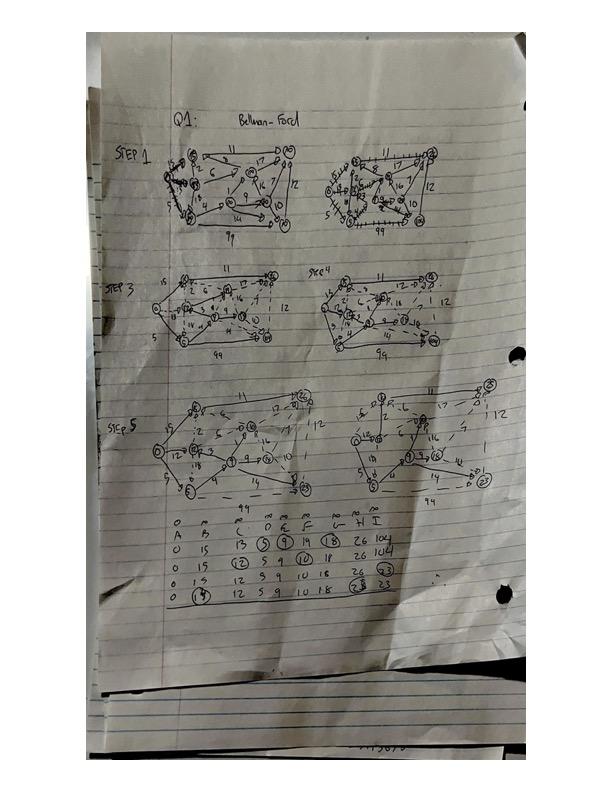
COMP 3804 Assignment 3

Nathaniel Mengistu

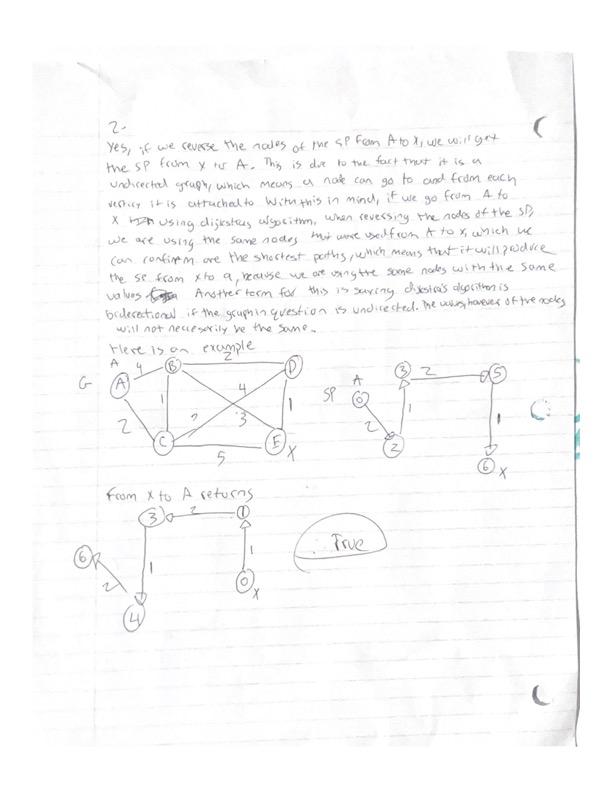
101155623

Jorge Sack



Final Tally

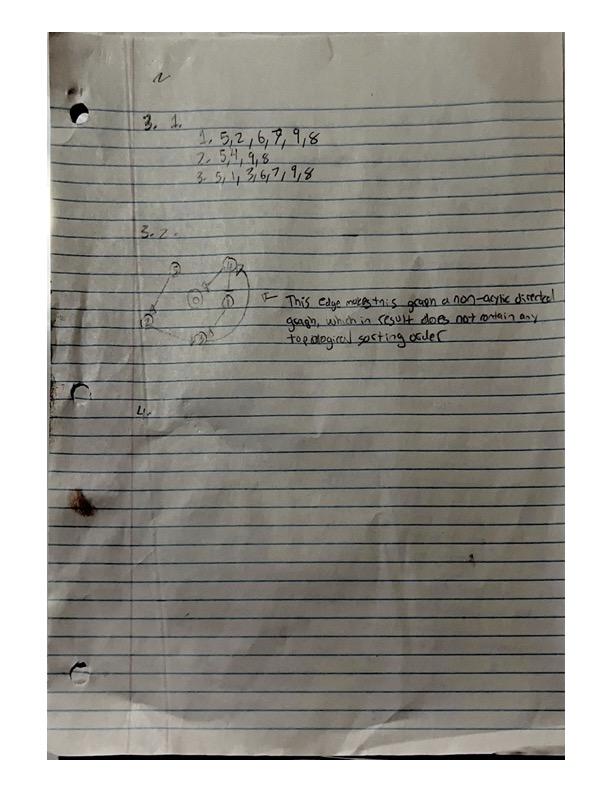
| A | B | C | D | E | F | G | H | I |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0/nil | 14/C | 12/E | 5/A | 9/D | 10/E | 18/E | 25/B | 23/E |



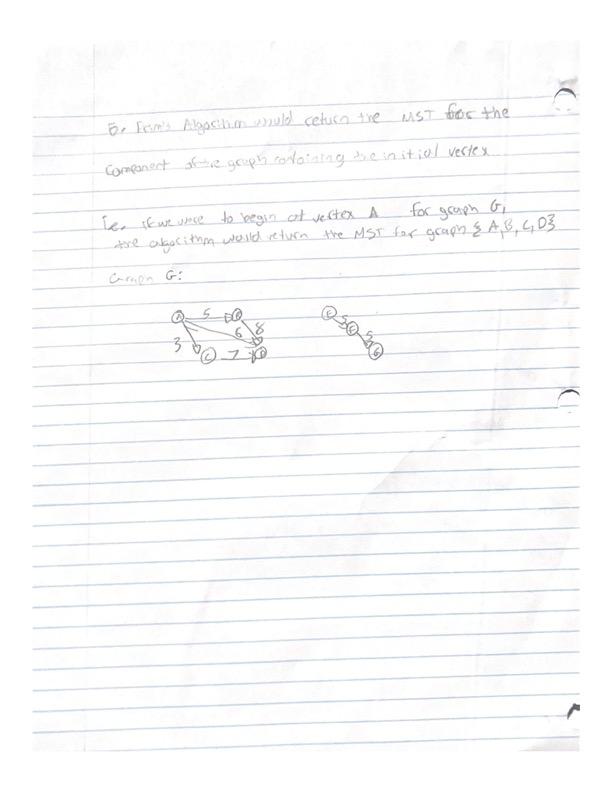
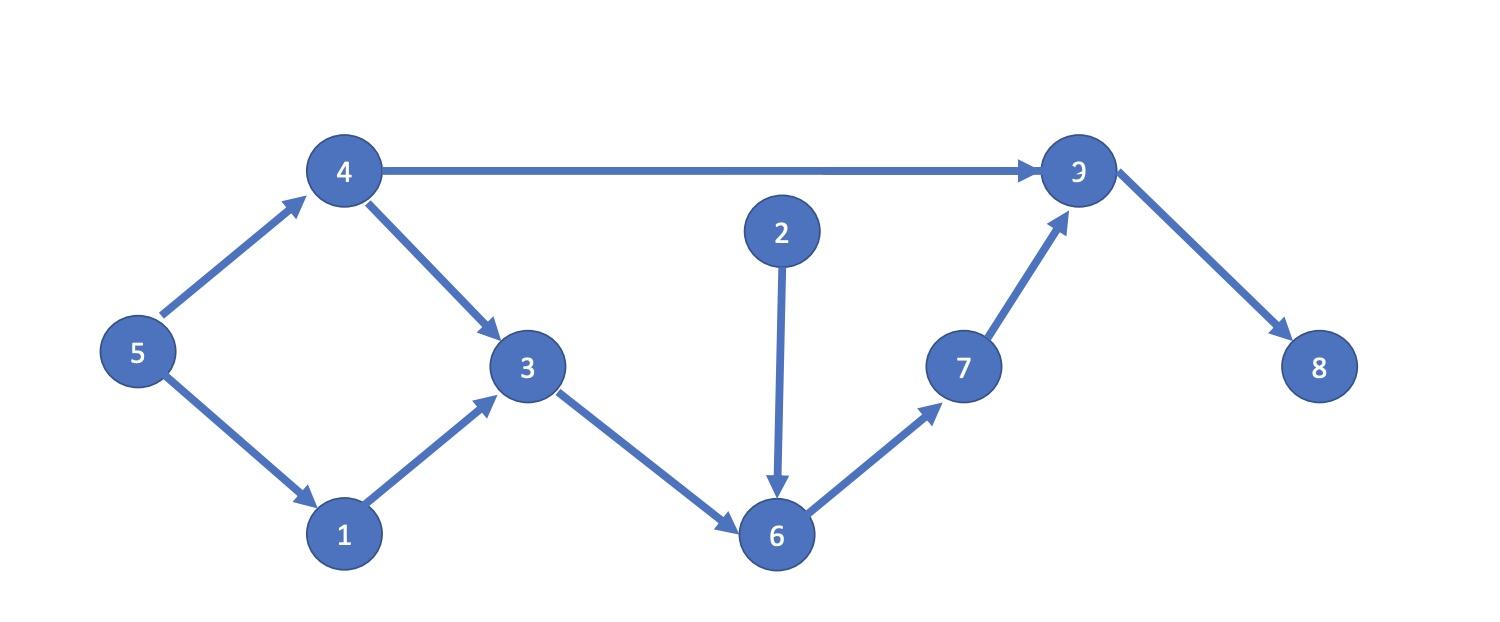
Q3.1Topological examples

1:512436798

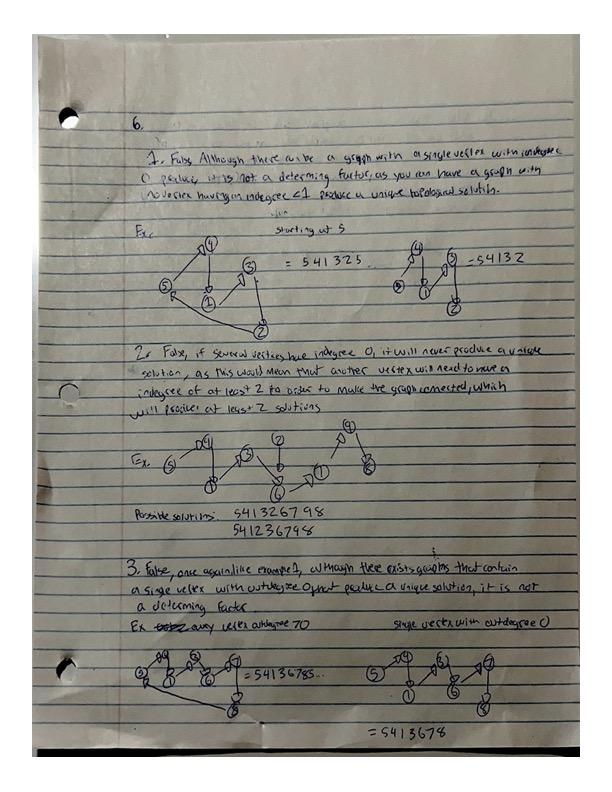
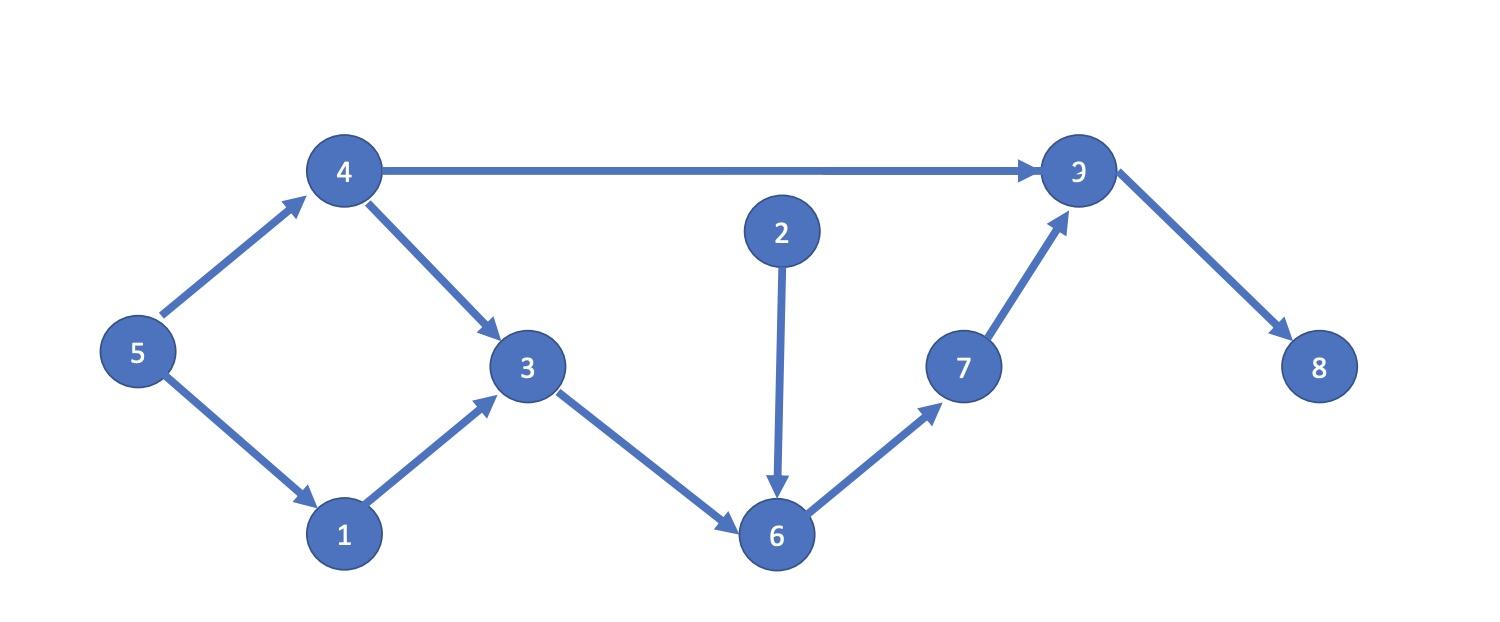
2:521436798

3:541326798Q4:**Can Dijkstra’s algorithm be used to determine if a graph is connected? How does that compare to algorithms for graph connectivity that you may know from previous classes or that you read up (and cite)? Just list the algorithm used and compare the time complexity.**

Dijkstra’s algorithm can be used to determine if a graph is connected, as the algorithm finds the shortest path from the source node to all other nodes within the graph, so if the return value of the algorithm does not contain every node in the graph, then we know that the graph is disconnected, as the resultant shortest path is from the source node to the last connected node on that section of the graph. Another algorithm that can be used to determine whether a graph is connected or not is a Breadth-First Search, or BFS of the graph. This algorithm will return all nodes in the graph. The runtime of BFS is O(V+E), where V is the number of nodes and E is the number of edges(via <https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/> ). The runtime for Dijkstra’s algorithm is O(V^2) (via <https://www.scaler.com/topics/data-structures/dijkstra-algorithm/> ), which is slower than BFS. This is because Dijkstra’s algorithm is comparing the weight of each verticy attached to the edge it is connected to before continuing on to a nother edge, whereas with BFS, it does not care about the weight of a verticy it will only check whether the edge is connected to a nother edge.

1:False, Although there exists graphs which contain a vertex with indegree 0 that provide a unique topological solution, it is not a determining factor, as a graph is only ever unique if it contains a hamiltonian path, and there are examples that will pass this statement but will not have a hamiltonian path thus deeming it not unique. An example would be the graph from q3:

As you can see 5 has a indegree of 0, yet it produces multiple topological solutions

3:False, Although there exists graphs which contain a vertex with outdegree 0 that provide a unique topological solution, like part 1 it is not a determining factor, as a graph is only ever unique if it contains a hamiltonian path, and there are examples that will pass this statement but will not have a hamiltonian path thus deeming it not unique. An example would be the graph from q3:

As you can see 8 has a outdegree of 0, yet it produces multiple topological solutions which proves my above statement

