Instructions: Please note that handwritten assignments **will not be graded**. Use the provided LaTeX template to complete your homework. Please do not alter the order or spacing of questions (keep each question on its own page). When you submit to Gradescope, you must mark which page(s) correspond to each question. **You may not receive credit for unmarked questions**.
When including graphical figures, we encourage the use of tools such as graphviz or packages like tikz for simple and complex figures. However, these may be handwritten only if they are neat and legible (as defined by the grader).

List any collaborators (besides TAs or professors) here:

1. (9 points) [W3, ⋆⋆] Topological Sort. For the following questions, select whether the statement is true or false, and write a *brief* (approximately one sentence) explanation of your reasoning.

   (a) The DFS-based topological sort algorithm we discussed in class uses vertex start times computed during a DFS traversal of the given DAG.

   □ True ■ False

   The DFS-based topological sort algorithm uses finish times, not start times, computed during a DFS traversal of the given Directed Acyclic Graph (DAG).

   (b) Topological Sort is $\mathcal{O}(E + V \log V)$, due to needing to sort the vertices by finish time (V is the number of vertices, E the number of edges).

   □ True ■ False

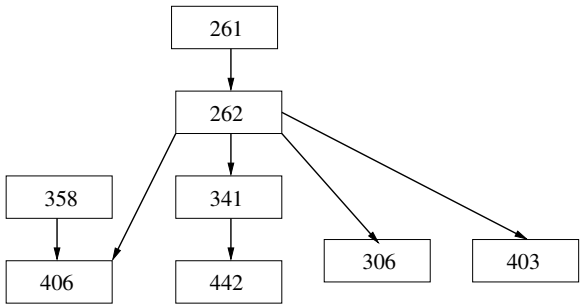
   The time complexity of the standard DFS-based topological sort is $\mathcal{O}(E + V)$, as it is a DFS traversal of the graph. The need for sorting the vertices is not a part of the standard topological sort algorithm.

   (c) After a topological sort has been performed, a topological order starts with a vertex that has no incoming edges (that is, in-degree of zero).

   ■ True □ False

   After a topological sort has been performed, the order starts with a vertex that has no incoming edges, meaning it has an in-degree of zero. This is a property of a valid topological order in a Directed Acyclic Graph (DAG).

2. (15 points) [W3, ⋆⋆] Perform toposort using a DFS



(a) Perform a DFS on the DAG in the figure above. Write the start and finish timestamps of each of the vertices.

Vertex	**Start Timestamp**	**Finish Timestamp**
261	1	14
262	2	13
306	9	10
341	5	8
358	15	16
403	11	12
406	3	4
442	6	7

(b) List the vertices in topological order based on the times above.

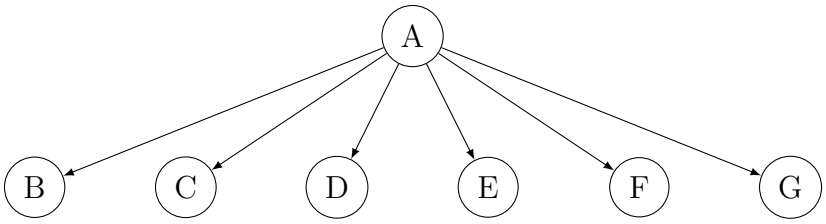
358, 261, 262, 403, 306, 341, 442, 406

(c) List an alternative topological ordering of the DAG above. (It does not have to be based on another DFS. Any valid ordering is acceptable.)

261, 358, 262, 403, 306, 341, 442, 406

I just swapped the first two vertices in the previous topological order.

3. (10 points) [W3, ⋆⋆] Draw a graph with 7 vertices and any number of edges such that there are 720 possible topological orderings. Provide a *brief* (approximately 2 sentence) explanation of your solution.



Explanation: For every possible topo ordering, the first node is 'A'. The remaining nodes can be in any order, so there are 6! = 720 possible orderings.

4. (9 points) [W4, ⋆⋆] Articulation Vertices. For the following questions, select whether the statement is true or false, and write a *brief* (approximately one sentence) explanation of your reasoning.

   (a) Consider unique vertices u, v, and x. If all paths between u and v include x, then x is an articulation vertex.

   ■ True □ False

   Removing x would disconnect all paths from u to v, so x is an articulation vertex.

   (b) An articulation vertex can have any degree ≥ 1.

   □ True ■ False

   An articulation vertex must have a degree ≥ 2 in order to disconnect the graph when removed. A vertex with degree 1 cannot disconnect the graph when removed, it would just remove a leaf node.

   (c) Consider a graph model of the internet where nodes are the various routers, servers, switches, devices, etc. that internet traffic can go through, and edges are connections between those network components. Assume all internet traffic from the Mines network goes through a single network switch in CTLM (there are no backup switches or alternative routes that internet traffic can take to exit the Mines network). The node representing that switch would be an articulation vertex in this model of the internet.

   ■ True □ False

   Removing the switch would disconnect the Mines network from the rest of the internet, so the switch is an articulation vertex.

5. (10 points) [W4, ⋆⋆] Strongly Connected Components. For the following questions, select whether the statement is true or false, and write a *brief* (approximately one sentence) explanation of your reasoning.

   (a) If a new edge is added to a directed graph, the number of strongly connected components may **increase**.

   □ True ■ False

   Can't increase the number of SCCs by adding an edge. Adding an edge can only decrease the number of SCCs or keep it the same.

   (b) If a new edge is added to a directed graph, the number of strongly connected components may **decrease**.

   ■ True □ False

   Adding an edge may decrease the number of strongly connected components. For example, adding an edge from c to b on the graph below would decrease the number of SCCs from 4 to 3.

   SCC's: $\{a, b, c, d, e\}, \{f, g\}, \{h\}$

   (c) If a new edge is added to a directed graph, the number of strongly connected components may **stay the same**.

   ■ True □ False

   Adding an edge may not change the number of strongly connected components. For example, adding an edge from b to c on the graph below would not change the number of SCCs.

   SCC's: $\{a, b, e\}, \{c, d\}, \{f, g\}, \{h\}$

Example:

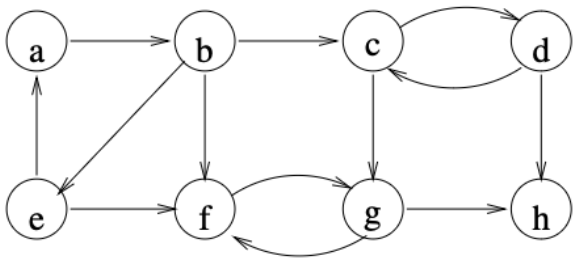


Figure 1: Simple directed graph with 3 strongly connected components.

SCC's: $\{a, b, e\}, \{c, d\}, \{f, g\}, \{h\}$

6. (20 points) [W4, ★★★] Answer the following questions about Strongly Connected Components.

   (a) (10 points) What is the minimum number of edges in an SCC with n vertices? Briefly describe how you would arrange the edges to create an SCC with n vertices and a minimal number of edges (we are looking for approximately one sentence).

   Min Edges: n

   Create a cycle with n vertices. Each vertex has an edge to the next vertex in the cycle. This creates an SCC with n vertices and n edges.

   (b) (10 points) Prove that the SCC component graph (the graph with SCCs condensed to single vertices with edges between SCCs if there are edges between vertices in the original directed graph) is always a DAG. (Looking for approximately 1 paragraph.)

   The SCC component graph is always a DAG because the original graph is a DAG. In the SCC component graph, each SCC is a single vertex. If there is an edge between two SCCs, it means there is a path between some vertex in the first SCC and some vertex in the second SCC. Since the original graph is a DAG, there cannot be a cycle in the SCC component graph (otherwise it would merge the SCCs). Therefore, the SCC component graph is always a DAG.

7. (12 points) [W5, ⋆⋆] MST. For the following questions, select whether the statement is true or false. **No explanation is necessary for these problems.**

   (a) Prim's and Kruskal's algorithms are both greedy algorithms.

   ■ True □ False

   (b) An MST of a graph is unique.

   □ True ■ False

   (c) In Kruskal's algorithm, it is correct to stop after adding $V - 1$ edges to the MST without considering the remaining edges.

   ■ True □ False

   (d) If an edge in the MST and another edge outside the MST have the same weight, replacing the edge inside the MST with the one outside yields another valid MST.

   □ True ■ False

8. (15 points) [AlgoBOWL, ⋆⋆] The following questions are about the AlgoBOWL project. For the following questions, select whether the statement is true or false. **No explanation is necessary for these problems.**

   (a) My team should expect to compute an optimal solution for every input with our algorithm.

   □ True ■ False

   (b) For an output from another team to be valid, it must match my team's output.

   □ True ■ False

   (c) The maximum number of edges in a prerequisite graph can be $n \cdot (n-1)$ for any n.

   ■ True □ False

   (d) Removing all vertices from the graph is a valid solution to the AlgoBOWL problem.

   □ True ■ False

   (e) If there are k groups in your AlgoBOWL section (including your group), your group will need to:

   □ Supply one input, compute k outputs and perform k verifications (not including open verification).
   ■ Supply one input, compute $k-1$ outputs and perform $k-1$ verifications (not including open verification).
   □ Supply one input, compute k outputs and perform k^2 verifications (not including open verification).
   □ Supply one input, compute one output and perform one verification (not including open verification).