Homework 15
CS 3385
Due April 21 at the end of class

1. Compilers often reorder source code for efficiency reasons. For example,

   ```
   x = 3
   y = 4
   z = x + y
   ```

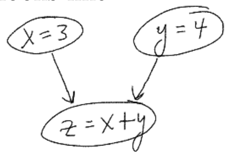
   could be reordered as

   ```
   y = 4
   x = 3
   z = x + y
   ```

   but *could not* be reordered as

   ```
   x = 3
   z = x + y
   y = 4
   ```

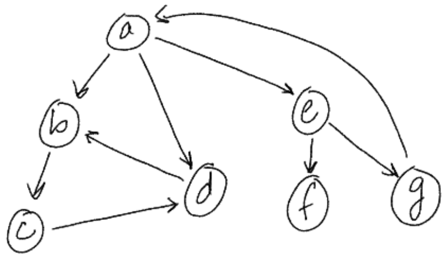
   because `z = x + y` has a dependency on `y = 4`, i.e., `y = 4` must be executed before `z = x + y` can be executed. One way to find a reordering of code is to build a dependency graph of the source code, and then do a topological sort on the graph. For example, the dependency graph of the above code looks like

   Given the code below, build a dependency graph with the vertices labeled with the line of code, and run a topological sort. Assume that the `for` loop of lines 5-7 of the `DFS` procedure considers the vertices in alphabetical order, and assume that each adjacency list is ordered alphabetically. Your answer will include both the graph labeled with discovery and finish times, as well as a new ordering for the code.

   ```
   a = 3
   b = a
   c = 4
   d = a
   e = d - a
   f = b + c + e
   ```

2. Show how the procedure `stronglyConnectedComponents` works on the following graph. Specifically, show the finishing times computed in line 1 and the forest produced in line 3. Assume that the loop of lines 5-7 of `DFS` considers vertices in alphabetical order and that the adjacency lists are in alphabetical order.

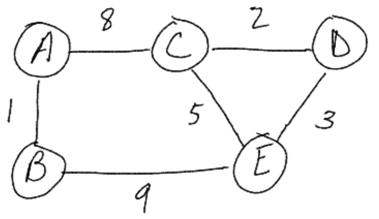
3. How can the number of strongly connected components of a graph change if a new edge is added?

4. Recall that a strongly connected graph is one in which there is a path (not necessarily direct) from any vertex to any other vertex. An Euler tour of a strongly connected, directed graph $G = (V, E)$ is a cycle that traverses each edge of G exactly once, although it may visit a vertex more than once. It turns out that G has an Euler tour if and only if `in-degree(v) = out-degree(v)` for each vertex $v \in V$. Give two examples of 4-vertex strongly connected, directed graphs: one that has an Euler tour and one that doesn't.

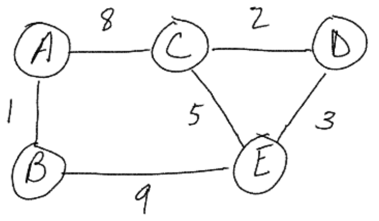
5. Let (u, v) be a minimum-weight edge in a connected graph G. Show that (uj, v) belongs to some minimum spanning tree of G.

6. Professor Notsosmart conjectures the following converse of Theorem 23.1. Let $G = (V, E)$ be a connected, undirected graph with a real-valued weight function w defined on E. Let A be a subset of E that is included in some minimum spanning tree for G, let $(S, V - S)$ be any cut of G that respects A, and let (u, v) be a safe edge for A that crosses $(S, V - S)$. Then, (u, v) is a light edge for the cut. Show that the professor's conjecture is incorrect by giving a graph that is a counterexample.

7. Run Kruskal's algorithm on the following graph. Show the table of processed edge and disjoint sets at each step using the format used in the video. Show the resulting minimum spanning tree.



8. Run Prim's algorithm on the following graph. Show the table of u, v and Q using the format used in the video. Start at A. Use alphabetical sorting for tie breakers in the priority queue. Show the resulting minimum spanning tree and include values of π for each vertex.


A
B
C
D
E
8
2
1
5
3
9