CS 163 — Winter 2024 — Midterm 1

Name:

UCInet id (your @uci.edu email address):

Use alternating seating.

Do not open this exam until told to start by the instructor.

This is a closed book, closed note exam. Calculators or other electronic devices are not allowed.

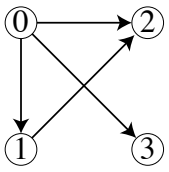
Please write your answers ONLY on the front side of each page.
Answers written elsewhere will not be scanned and will not graded.

Restroom breaks are not allowed: If you leave the classroom, you must turn in your exam first.

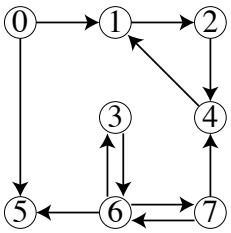
You may use the back sides of the pages as scratch paper.
Do not unstaple the pages.

1. (20 points) (a) Draw the directed graph whose Python representation is
   { `0: [1,2], 1: [2,3], 2: [3,0], 3: [2]` }.

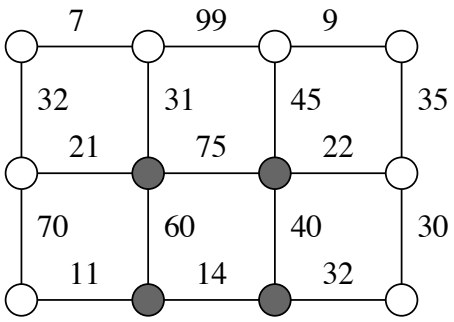
   (b) Write down the Python representation for the directed graph shown below.



2. (20 points) List the strongly connected components of the directed graph shown below.



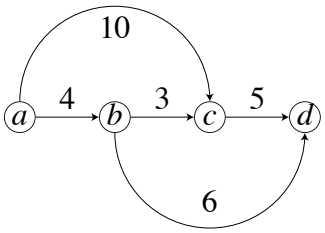
3. (20 points) In the graph shown below, the four dark shaded vertices form a cycle.



(a) If we apply the cycle rule to this cycle, what does it tell us about the MST of the graph?

(b) If we apply the cut rule to the cut that separates the four shaded vertices from the rest of the graph, what does it tell us about the MST of the graph?

4. (20 points) Suppose that we use longest-path scheduling on the activity-on-edge graph shown below.



(a) What is the critical path in this graph?

(b) At what time is each milestone of the project represented by this graph scheduled? (You may assume that the edge weights are in units of days.)

5. (20 points) State, using O-notation, for a graph with n vertices and m edges:

   (a) The time bound for finding single-source shortest paths in a DAG, using relaxation in a topological order

   (b) The time bound for finding single-source shortest paths in a directed graph with positive edge weights, using Dijkstra's algorithm with Fibonacci heaps

   (c) The time bound for finding single-source shortest paths in a directed graph with negative edge weights but no negative cycles, using the Bellman–Ford algorithm

   (d) The time bound for finding all-pairs shortest paths in a directed graph with negative edge weights but no negative cycles, using Johnson's algorithm with Fibonacci heaps