Name: Jae Park RCS ID: Parkj23 @rpi.edu

♡ ♡ CSCI 2300 — Introduction to Algorithms ♡ ♡
Fall 2020 Exam 1 (February 20, 2020)

- Please silence and put away all laptops, phones, calculators, electronic devices, etc.
- You may use your printed notes and book(s) for this exam
- This exam is designed to take 100 minutes, but we will use the full 110 minutes from 6:00-7:50PM; for 50% extra time, the expected time is 150 minutes, i.e., 5:00-7:30PM
- During the exam, **questions will not be answered** except when there is a glaring mistake or ambiguity in the statement of a question; we cannot clarify a question for you; please do your best to interpret and answer each question clearly and concisely
- Long answers are difficult to grade; the space provided should be sufficient for each question; however, you may use the last page of this exam for overflow work
- All work on this exam must be your own; do not even think of copying from others
- When you hand in your exam, be prepared to show your RPI ID

Please sign below to indicate that you will not copy or cheat on this exam:

Signature: [illegible]

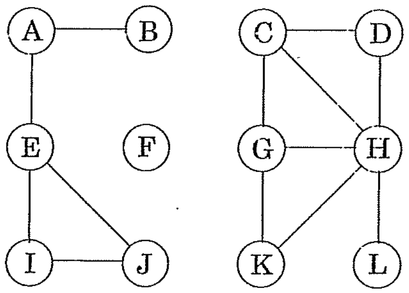
Do not start this exam until you are instructed to do so.

1. **(3 POINTS)** Given undirected graph $G = (V, E)$ represented by an adjacency matrix, what is the runtime of DFS to determine whether node $t \in V$ is reachable from node $s \in V$? Assume individual lookups in the adjacency matrix are $O(1)$. Clearly circle the best answer.

   (a) $O(|V|^2)$
   (b) $O(|E|^2)$
   (c) $O(|V|)$
   [✓ (d) $O(|E|)$]
   (e) $O(|V| + |E|)$ √
   (f) $O(|V| + |E|^2)$

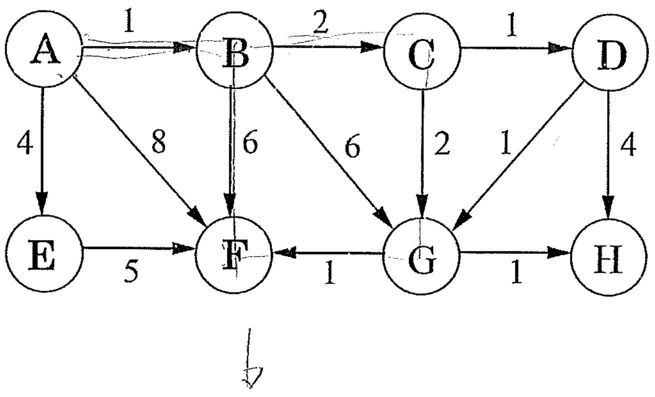
2. **(3 POINTS)** How many connected components are there in the undirected graph below? Clearly circle the best answer.

   (a) 0
   (b) 2
   (c) 4
   (d) 1
   [✓ (e) 3]
   (f) 5



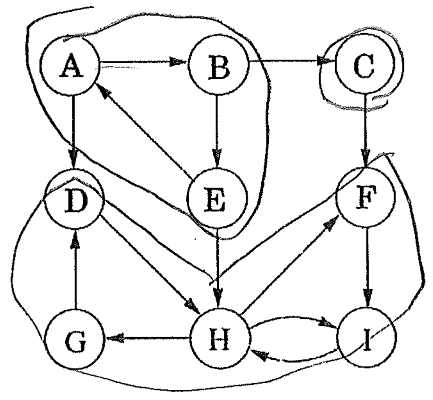
3. **(3 POINTS)** Applying Dijkstra's algorithm to the directed graph below, what is the shortest distance (i.e., minimum sum of all edge weights) from node A to node F? Clearly circle the best answer.

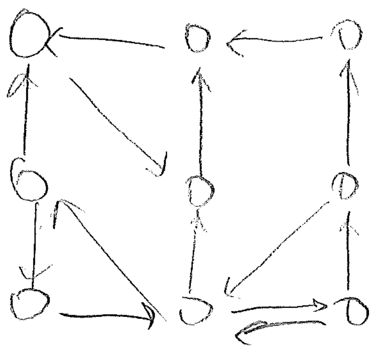
   (a) 9
   (b) 8
   (c) 7
   [✓ (d) 6]
   (e) 5
   (f) Node F is unreachable from node A since all nodes must be visited at least once



4. **(3 POINTS)** How many strongly connected components are there in the directed graph below? Clearly circle the best answer.

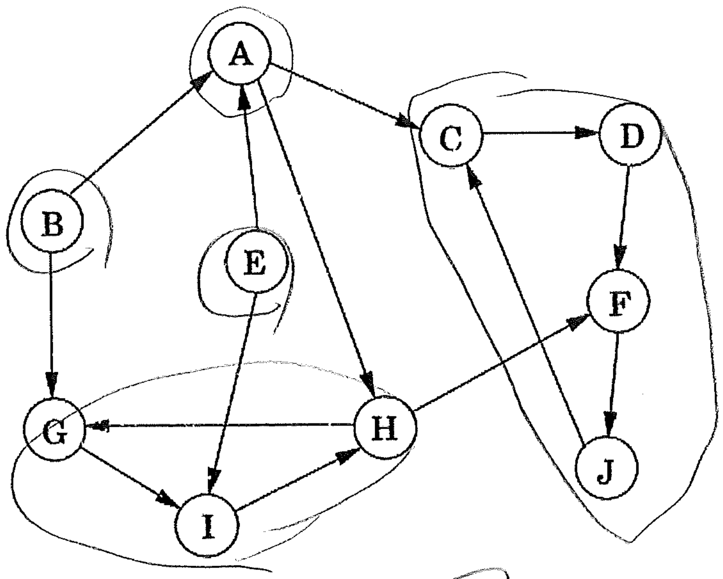
(a) 0
(b) 2
(c) 4
(d) 1
[✓ (e) 3]
(f) 5

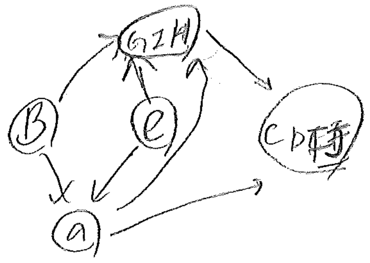




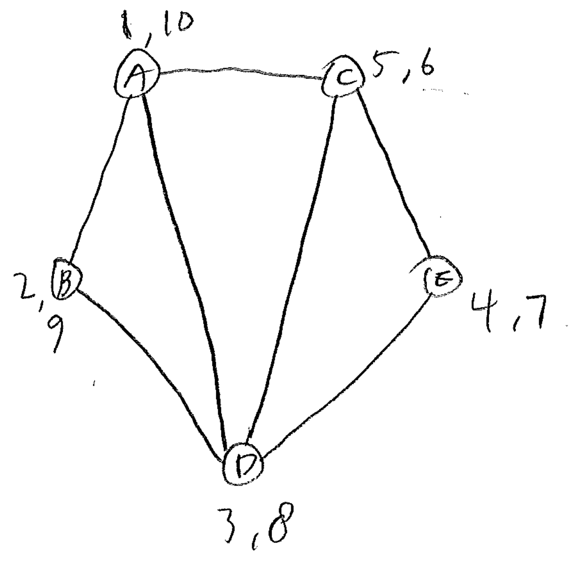
5. **(3 POINTS)** What is the minimum number of edges you must add to the directed graph below to make it strongly connected? Clearly circle the best answer.

(a) 0
[✓ (b) 2]
(c) 4
(d) 1
(e) 3
(f) 5

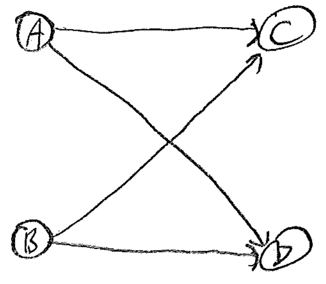




6. **(12 POINTS)** Draw an undirected graph G with five nodes and seven edges such that the pre and post numbers from the DFS algorithm for all but one of the nodes differ by at least 3 (i.e., for each node u in G, $\mathtt{post}(u) > \mathtt{pre}(u) + 2$).



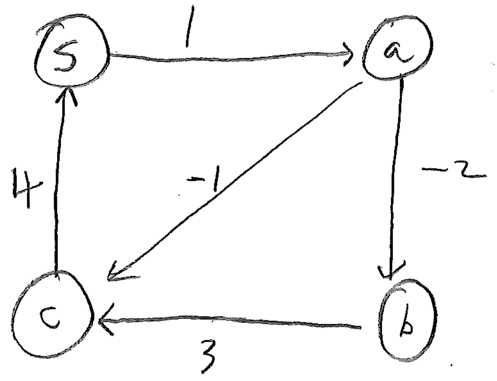
7. **(12 POINTS)** Draw a directed acyclic graph (DAG) with four nodes that has two sources and four distinct topological orderings.



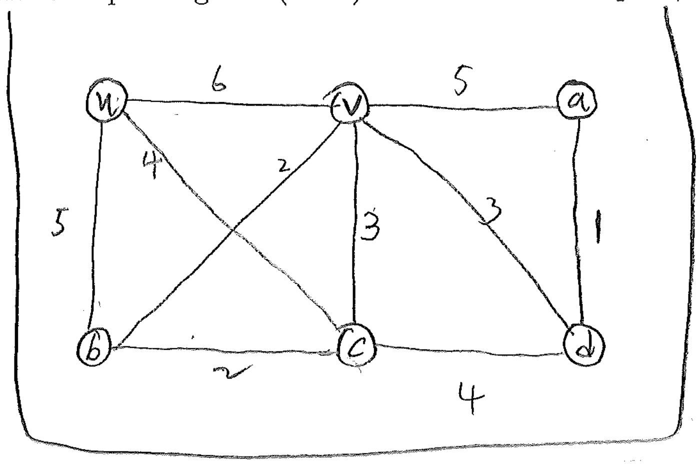
A B C D
A B D C
B A C D
B A D C

8. **(12 POINTS)** Draw a graph with ~~five~~ 4 nodes for which Dijkstra's algorithm fails to find the shortest path~~, but the~~ Bellman-Ford algorithm succeeds. ~~Further, make sure the shortest path is unique (i.e., exactly one shortest path).~~

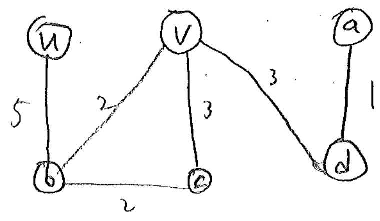
5



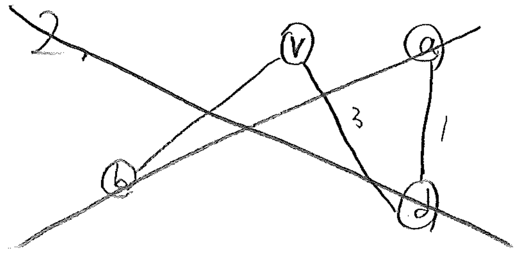
9. **(12 POINTS)** Draw a connected undirected graph with six nodes and at least six edges in which the shortest (i.e., minimum weight) path between two nodes u and v is **not** part of any minimum spanning tree (MST). Show the shortest path, then draw all possible MSTs.



1.

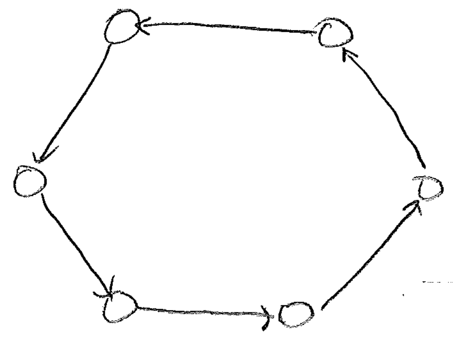


~~2.~~



10. **(12 POINTS)** Draw a strongly connected directed graph $G = (V, E)$ with $|V| = 6$ such that, for every $u \in V$, removing u from G leaves a directed graph that is no longer strongly connected.

$< G >$



11. **(12 POINTS)** Write an algorithm to find a path that traverses all edges of directed graph G exactly once or determines that such a path does not exist for G. You may visit nodes multiple times, if necessary. Show the runtime complexity of your algorithm.

We will look at indegree and outdegree values for all the nodes.

1. For all but two nodes : $u \in V$
   indegree (u) = outdegree (u)

2. For the two exceptional nodes : $v, w \in V$
   outdegree (v) = indegree $(v) + 1$
   indegree (w) = outdegree $(w) + 1$

$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \end{array}$$

Assume you are using adjacency matrix.
The algorithm will go through the matrix $|V|$ times (for each node) and calculate the values of indegree and outdegree. In order to calculate the indegree of a vertex, you add up all the numbers in the corresponding column $(O(|V| \times 1) = O(|V|))$, and for outdegree, you sum the corresponding row$(O(|V|))$. Calculating indegree and outdegree for each vertex takes $O(2|V|) = O(|V|)$, and you do that for every vertex. So $O(|V| \times |V|) = O(|V|^2)$. We can store all the indegree and outdegree values in arrays indegree[], outdegree[]. Now we iterate through the two arrays and use the algorithm above. If the algorithm doesn't pass, then it means that there doesn't exist

12. **(13 POINTS)** Consider the following pseudocode of a function that takes integer $n \geq 0$ as input.

```
function netflix(n):
  print '*'
  if n == 0: return
  for i = 0 to n - 1:
    print '*'
  netflix(n - 1)
  return
```

0 * (0)

1 * * * (1)

2 * * * * * * (3)

3 * * * * 10 (6)

3, 4

$\frac{n(n+1)}{2} + (n+1)$

$2n+2$

n^2+3n

$\frac{(n+1)(n+1)}{2}$

Let $T(n)$ be the number of times the above function prints a star (`'*'`) when called with valid input $n \geq 0$. What is $T(n)$ exactly, in terms of n only (i.e., remove any reference of $T()$ on the right-hand side). Prove your statement.

By observing the pattern of netflix(), we can identify that $T(n) = (n+1) + T(n-1)$. Expanding out a few terms tells us that it is simply the sum of integers from 1 to n plus (n+1) number of stars being printed out by the first line of the code.

Therefore, $T(n) = \sum_{i=0}^{n} i + (n+1) = \frac{n(n+1)}{2} + (n+1)$

$= \frac{n^2+n+2n+2}{2}$

$= \frac{(n+2)(n+1)}{2}$.

The $\{\sum_{i=0}^{n} i\}$ part is proven by the definition of literal summation (Gaussian).

Use this page for any scratch or overflow work.