Exam 3 CS-GY 6033 CF01 Fall 2023
Dec 21st 2023

Instructions:

Scheduling:

- The exam runs from 6pm to 9:00pm on Dec 21tst 2023. Your exam is run through Gradescope, and will available to download at 5:50pm. The exam is to be completed by 9:00pm. Gradescope stops accepting uploads at 9:15pm. There are absolutely no late uploads accepted by the system after that time. The time to complete the exam will depend on your preparation: a prepared student could finish it in less than two hours, an ill-prepared student might take up to three hours. It is your responsibility to allow time for uploading your exam.

Format:

- The exam consists of a total of 100 points. Your online quiz is worth 20 points.
- **Submission Penalty:** if you do not submit your exam on Gradescope, or do not properly assign your pages, you receive a deduction of 3 points on your exam.
- You may write your solutions directly on the downloaded exam paper, *or* in your own format. You are responsible for providing clear and legible solutions to the problems. Your exam must be resubmitted into Gradescope electronically. Ensure that you know how to quickly upload any handwritten material. This is entirely the student's responsibility. You may assign your pages after the deadline.

Questions during the exam:

- You will ask questions directly by email.

Rules:

- This exam is a **take-home exam**. You may use **only** the resources from the online class (any material on NYU classes for this course) and any type of calculator (although it is not needed).
- Your work must be entirely your own. It is **forbidden to discuss any work with *any* other person**. Furthermore, your work must be done without using internet searches (although this is completely unhelpful for this exam). Any breach of academic honesty will be handled in accordance with the *Student Code of Conduct*, (a copy of which is provided), and in this particular case, taken very seriously.
- You are asked to **read** the attached Student Code of Conduct Section III subsections A,B,C,D,E and **sign** below to acknowledge that you aware of the policy. Once signed, a copy of this page must be uploaded with your exam.

I acknowledge that my submitted Exam work is entirely my own. I have read and am in accordance with the Student Code of Conduct policy of NYU Tandon and fully accept the consequences of breaching the above instructions.

Name: ____________________

Signature: ____________________

Instructions

Read the following questions carefully. Recall that you may use algorithms from class. For example, you may reference BFS, Dijkstra's algorthtm, etc, as long as you carefully describe the input parameters.

Your written exam has a total of 80 points. Your oral question is out of 20.

YOU WILL ASK YOUR QUESTIONS DIRECTLY BY EMAIL. You do NOT need to join a ZOOM meeting. I will be connected the entire time.

REMINDER of SUBMISSION PENALTY: if you do not submit your exam on Gradescope, or do not properly assign your pages, you receive a deduction of 3 points on your exam.

Question 1

6 points

Let G be a directed, unweighted graph with exactly k strongly connected components. If a single directed edge is added to G (where there was not already an edge), which of the following is possible? There may be more than one possibility.

1. The number of strongly connected components *increases*
2. The number of strongly connected components *decreases*
3. The number of strongly connected components *stays the same*

Justify your answer.

Question 2

8 points

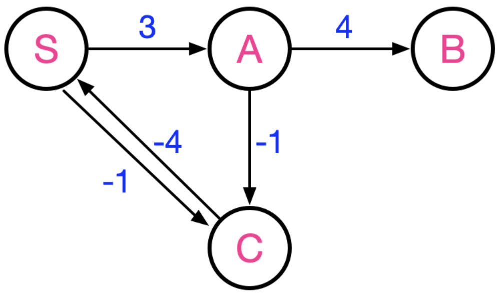
G is a directed, weighted graph (shown below).

- Execute Bellman-Ford's algorithm (to completion) using vertex S as the source, where the edges are processed in the following order

$$SA, SC, AB, AC, CS$$

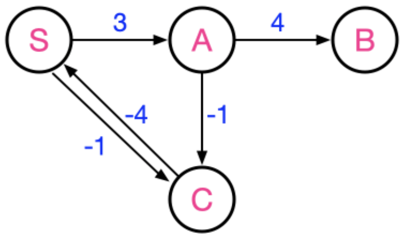
You must show how when a distance attribute and parent attribute are updated, and list when any edges result in no changes. **You must show the current edges of the SSSP tree**.

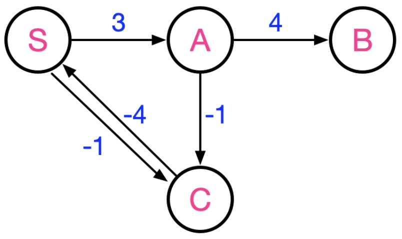
**you may use the next page if you like, to trace out the steps of BF on the graph*

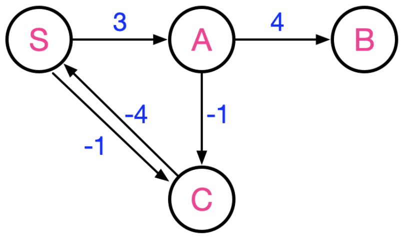


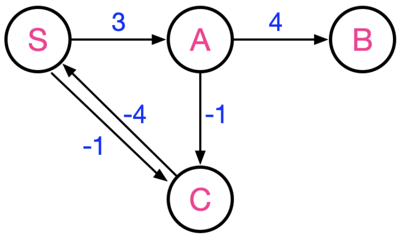
- The value of $C.parent$ is updated **how many times in total** before the BF algorithm terminates?

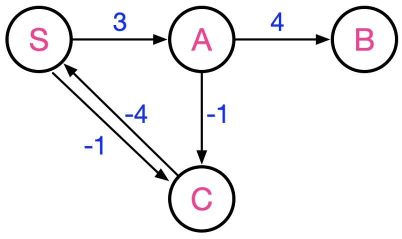
Use this sheet if you like. If you use this sheet, you must clearly label your steps and show your distance attributes as they are updated.

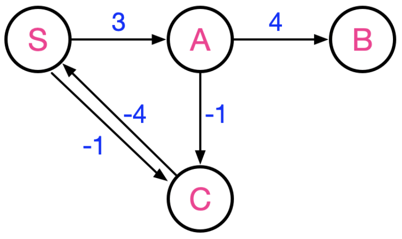


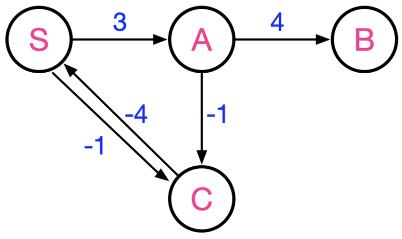


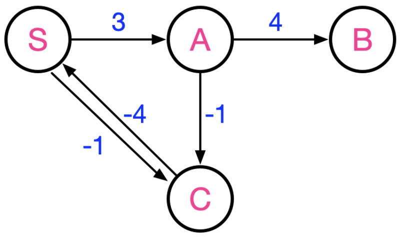


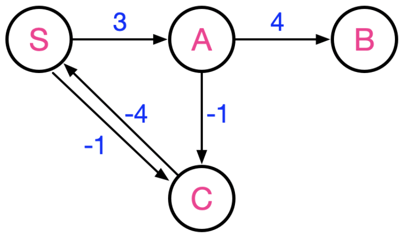


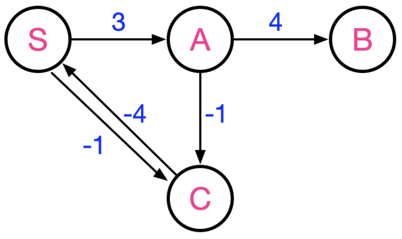


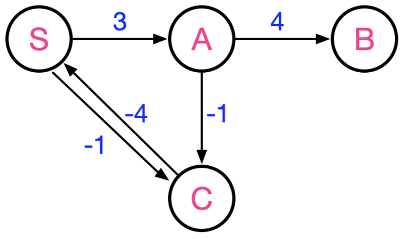


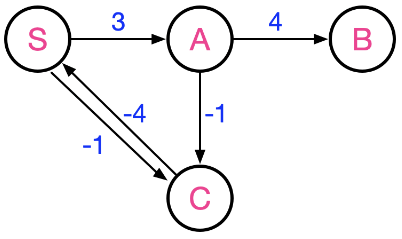












Question 3:

8 points

Let G be a **directed, unweighted** graph with vertex set V and edge set E. Suppose that each edge e has additional attribute *e.color*, which is either *red* or *green.* A particular vertex is labelled N for the north pole, and a particular vertex is labelled S for the south pole. The goal is to determine if there is a route from the south pole to the north pole, which consists of edges of **all one color**, OR, **starts on edges of one color, and then *switches* to edges of another color**. For example, an all-red path is acceptable, or a path that starts with red edges, and then switches (once) to green edges, or a path that starts with green edges and then switches along the way (once) to red edges, or an all-green path.

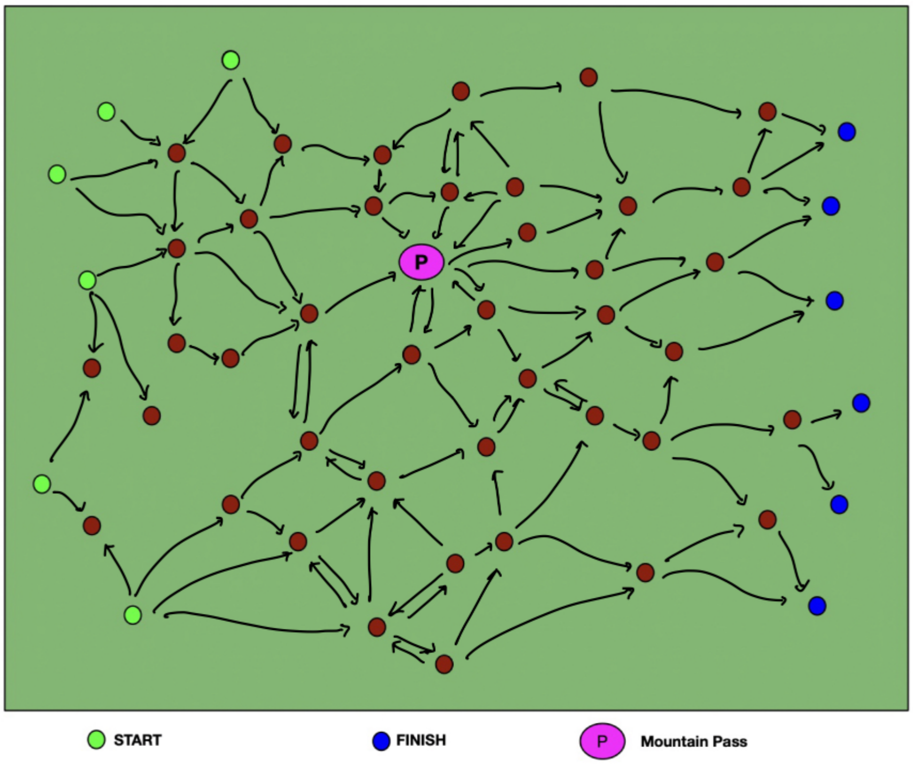
Your job is to design an algorithm that solves this problem. You do not need to provide pseudo-code, but you must carefully described the steps and the inputs to any algorithms from class that you use. Justify the runtime of $O(E + V)$.

Question 4

10 points

Recall the mountain biking problem from class. In this updated version, each of the mountain bike trails can be taken in **only one direction**. Furthermore, there is only **one** mountain pass, called P. There are n marked routes on the map, of which $n/5$ are starting positions, and $n/5$ are finishing positions. There are m direct routes on the map, where each direct route between two points has both a direction (possibly bi-directional) and a distance.

Your job is to determine the shortest distance through the mountain from a start point, to the mountain pass, to a finish point. Be sure to consider all possible starting and ending position. You must carefully describe the graph model to your problem, and describe your steps. Be sure to describe the input and output to any algorithm used from class. The algorithm must find the shortest distance from a start to a finish point, and also a print out of the stops from start to finish (in order). Justify the runtime of $O(m \log n)$, where n is the number of trail points and m is the number of bike trails.



Question 5

10 points

Let G be a **directed, weighted** graph, where each edge weight is a positive integer. Let s and t be particular vertices in the graph. Suppose we attempt to update Dijkstra's algorithm, so that it finds the path of **maximum total weight** path from vertex s to t. Show how to update the pseudo-code of Dijkstra's algorithm so that it uses a *max* priority queue. Ensure that you update all parts of the pseudo-code so that the algorithm is greedily selecting maximum routes.

Does this updated version correctly find the maximum path from s to t on any graph G? Justify your answer by either explaining *why* or providing an example where it fails.

Question 6

8 points

Let G be an undirected, connected graph, where each vertex has an additional attribute called $v.color$, which may be either black or white. Your job is to update DFS-visit(u) so that it counts the total number of black nodes in the graph. The procedure must return the total number of black nodes. Justify the runtime of $O(V + E)$.

Question 7

10 points

In class we saw an algorithm for finding a topological sort of the vertices in a directed, acylic graph. The algorithm was based on finding $v.indgree$ for each vertex in the graph, and it used an external list L. Suppose we wish to update this algorithm in order to solve the following problem:

> Let G be a directed, acyclic graph, where each vertex has a difficulty rating, $v.rate$. The problem is to find a topological ordering of the vertices, one which prioritizes vertices of *low* rating before vertices of *high* rating. For example, if several orderings are possible, the goal is to prioritize vertices of *lowest rating* before those of *higher rating*, as long as it respects the topological sort.

Question 8

12 points

For each problem below, determine if there is a known polynomial-time algorithm for solving the problem. You must justify your answer using problems and algorithms defined in this course.

1. Graph G is a simple, connected, undirected graph. There are n vertices and m edges. The goal is to color at least k vertices pink, and the remaining $n - k$ vertices in green, in such a way that no two pink vertices share an edge. Note that it is possible for two green vertices to share an edge.
2. Graph G is a simple, connected, undirected graph. There are n vertices and m edges The goal is to find at least k vertices that can be colored pink, and the remaining $n - k$ vertices that can be colored blue, in such a way that no two pink vertices share an edge, **and** no two blue vertices share an edge. Note that each vertex must be colored either pink or blue.
3. Graph G is a simple, connected, undirected graph, where each vertex is already *assigned* one of two colors: pink or blue. The goal is to determine if there is a cycle consisting of only pink vertices.

Question 9

8 points

Let G be a **directed, weighted** graph on n vertices and m edges. A particular vertex is marked as the source vertex s. Consider the problem of deciding if the graph contains a path of total weight **at least** k, starting from vertex s.

Show that this problem is NP-complete. You must use a reduction from one of: TSP, Hamiltonian Path, Subset Sum, Clique.