COMP 285 (NC A&T, Spr '22) Homework 6

Due. Wednesday, March 23rd, 2022 @ 11:59 PM!

Homework Expectations: Please see Homework.

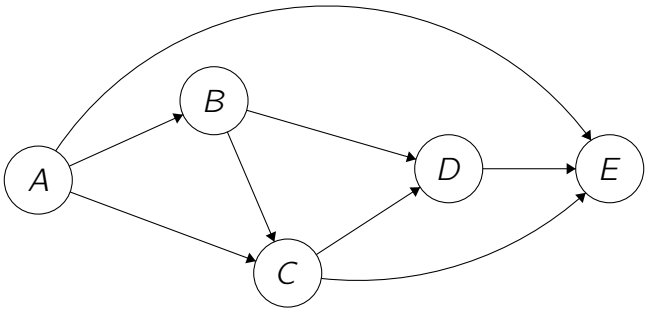
Exercises The following questions are exercises. We encourage you to work with a group and discuss solutions to make sure you understand the material.
Points This assignment is graded out of 100 points. However, you can get up to 120 points if you complete everything. These are not bonus points, but rather points to help make-up any parts you miss.

Fun with Graphs

Written Problems The following questions are to be submitted in written/typed form to gradescope.

1 Exercise: DFS Basics (12 pt.)

Consider the following directed acyclic graph (DAG):



In class, we saw how to use DFS to find a topological ordering of the the vertices; in the graph above, the unique topological ordering is A, B, C, D, E. We saw an example where we happened to start DFS from the first vertex in the topological order. In this exercise we'll see what happens when we start at a different vertex. Recall that when you run DFS, if it reached a node with no children (i.e. can't go any further), then it will resume the search at an unvisited vertex.

1.1 (6 pt.)

Run DFS starting at vertex C, breaking any ties by alphabetical order.[1]

(a) **(3 pt.)** What do you get when you order the vertices by **ascending** start time?

(b) **(3 pt.)** What do you get when you order the vertices by **descending** finish time?

1.2 (6 pt.)

Run DFS starting at vertex C, breaking any ties by **reverse** alphabetical order.[2]

(a) **(3 pt.)** What do you get when you order the vertices by **ascending** start time?

(b) **(3 pt.)** What do you get when you order the vertices by **descending** finish time?

[We are expecting: For all four questions, an ordering of vertices. No justification is required.**]**

[1] For example, if DFS has a choice between B or C, it will always choose B. This includes when DFS is starting a new tree in the DFS forest.

[2] For example, when DFS has a choice between B or C, it will always choose C. This includes when DFS is starting a new tree in the DFS forest.

2 Exercise: The World as Graphs (12 pt.)

For each of the following real-world problems, try to formulate this as a problem about graphs.

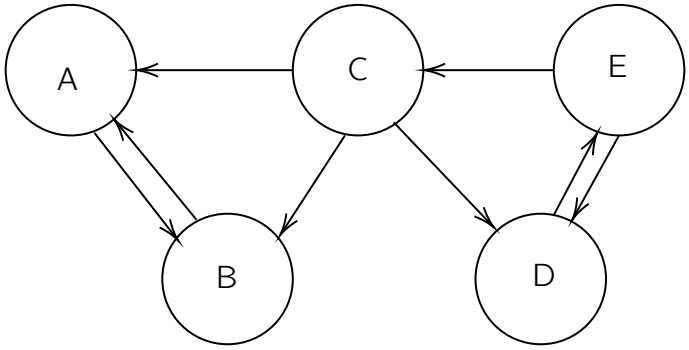
Recall a graph is just a collection of vertices and edges.

What algorithmic problem about graphs do we need to solve in order to solve the following problems? (Note, you don't actually have to solve these problems, just transform them into graph problems that we might then be able to solve).

1. **(4 pt.)** Among actors, a "Bacon number" is the number of degrees of separation from an actor to Kevin Bacon. For example, Kevin Bacon's Bacon number is 0. If an actor works in a movie with Kevin Bacon, the actor's Bacon number is 1. If an actor A works with an actor B who worked with Kevin Bacon in a movie, then actor A's Bacon number is 2, and so forth. How would you represent this as a graph so we could answer questions like: (a) What is Samuel L. Jackson's Bacon number? and (b) list all the people with Bacon number equal to 6.
2. **(4 pt.)** You need to take a bunch of classes at North Carolina A&T University, and some of them depend on each other. For example, you must take COMP 280 before taking COMP 285.Given a set of classes you need to take, and information about which class is a pre-requisite for which other class, generate an order in which to take all of the classes. Assume you can only take one class at a time. How would you repesent this as a graph so we can answer this question?
3. **(4 pt.)** You are about to purchase a bunch of fish. You have two very large fishtanks. Unfortunately, some of these species of fish will fight if they are put in the same tank. For each pair of species, you know whether they will fight or whether they will peacefully co-exist. How would you represent this as a graph such that you can find a way to separate the fish into two peaceful fishtanks if it exists.

[We are expecting: For each of the three questions above, explain how you'd represent them as graphs. What are the nodes? What are the edges? Are the edges directed? Are they weighted?**]**

3 Exercise: Kosaraju's algorithm (16 pt.)



In this question, we ask you to run Kosarju's SCC algorithm (see Lecture 18, Slide 45). We summarize the gist here:

When running DFS, we will **break any ties by alphabetical order**. Remember that if DFS has reached a node with no un-visited children (i.e. can't go any further), then it will resume the search at an unvisited vertex. We will decide which unexplored vertex to continue DFS by **alphabetical order**.

3.1 First DFS (4 pt.)

What are the start and finish times of each node after running the first DFS from node *C* on the **original** graph?

Node	Start Time	Finish Time
A		
B		
C	0	
D		
E		

[We are expecting: Fill in this table**]**

3.2 Second DFS (6 pt.)

Next, we will run DFS from the node with the largest finish time on the **reversed** graph. What do you get when you order the vertices by **ascending** start time for the second DFS run?

[We are expecting: An ordering of vertices. No justification is required**]**

3.3 Connected components (6 pt.)

What are the strongly connected component(s) in this graph?

[We are expecting: A list of sets of nodes. No justification is required**]**

4 Most Reliable Path (20 pt.)

You are planning for an exciting trip from the nice little town s to the shining new city t this summer! Sadly there is no direct flight from s to t so you need to take multiple flights to get to t. You created a directed graph $G(V, E)$ where each node represents an airport and each edge represents a flight connection from one airport to another.

Each flight connects two airports, but for each flight e there is some probability p_e that this flight can get cancelled due to COVID-19. Each p_e is some value between 0 and 1, where 0 means that this flight will always take place, and 1 means that this flight will always get cancelled. You have to book your flights and hotels in advance. You can assume that all p_e are independent of each other and **the graph has no cycles**.

4.1 Measure of Reliability (5 pt.)

We can measure the reliability of a path P as the probability having all flights in P not being cancelled (eg, all the flights succeed). Suppose you decide to take the flights along the path P with k flights given by $e_1 \to e_2 \to \cdots \to e_k$. How reliable is this path, using $p_{e_1}, p_{e_2}, \cdots, p_{e_k}$.[3].

[We are expecting: A mathematical expression summarizing the reliability of the path P.**]**

4.2 Just to be safe? (15 pt.)

Create an algorithm to find the most reliable path between s and t. By most reliable path, we mean that the path P you find from s to t should have the highest probability of having all flights in P not being cancelled. Having plenty of time for the trip, you **DO NOT** care about how long or how much it takes you to travel, but you hope to reach the destination with **exactly the route you planned**.

[Hint: Recall that $\log(A \times B) = \log(A) + \log(B)$, so you can convert multiplication into addition. It might help to recall Dijstra's Algorithm (Lecture 22). Lastly, note that $\log(x)$ where $0 \leq x \leq 1$ is negative, with smaller values of x giving more negative results.]

[We are expecting: Pseudocode or a very clear English description of your algorithm, an informal justification that your algorithm is correct, and brief analysis of running time**]**.

4.3 Ethics (5 pt.)

Imagine that, based on the excellent results of your flight planning algorithm, you have been hired by an online travel agency to give your expert advice on their route recommendation strategies. You discover that, by maximizing the reliability of suggested routes, the company

[3]For example, say I had a path that took flights e_1, e_2, and e_3. Then this whole trip succeeds only if e_1 **and** e_2 **and** e_3 are **not** cancelled. Since the events are independent, the reliability is given by $(1-p_{e_1})(1-p_{e_2})(1-p_{e_3})$

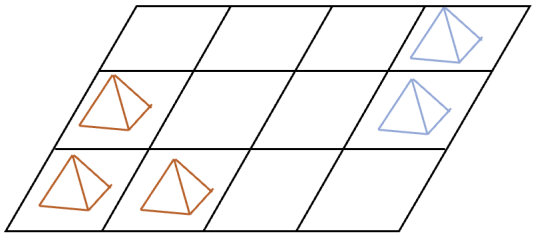
would be overlooking other considerations such as carbon and time efficiency. Write a one-paragraph memo to the programming team explaining why these (or other) values should be taken into account.

[We are expecting: two to four sentences explaining (1) what other measures are overlooked by an algorithm that maximizes reliability; and (2) the reasons why the airline should care about these measures, including some potential consequences of the omission**]**

5 Summer Camp(10 pt.)

Some North Carolina A&T bulldogs decided to hold a summer camp event at the Hanging Rock State Park. The camping ground is divided into an $m \times n$ grid, where each cell can accommodate one standard-sized tent. The North Carolina A&T bulldogs put up their tents connected on the grid (considering four cardinal directions). The following day, they discover another connected collection of tents put up by bulldogs from UNCG. Two groups of bulldogs are willing to make friends with each other and connect their living places into a single collection by putting up some spare tents. Please help design an $O(mn)$ algorithm that calculates the **minimum number of spare tents** that the bulldogs need to put up.

Example of camping ground layout If run your algorithm on this, the output should be 2 (We need to put up at least 2 tents to connect the two groups of tents).



Input: A nested array A of $m \times n$ where 1 represents a tent and 0 represents an empty camping ground.

[hint: You might want to locate all tents in one connected collection of tent first.]

[We are expecting: Pseudocode of the algorithm or a clear English description, **the algorithm should run in O(mn) time.]**

Coding Problems The following questions are to be submitted as a ".zip" file on Gradescope.

6 Coding (50 pt.)

After completing the written portion of the assignment, you should submit it to Gradescope.

For the coding portion, get your starter C++ code or Python code.

Note that the starter code also include a few test cases you can run on repl.it. However, the full test suite is the one run on Gradescope.

Please reference the `README.md` included in your starter code for detailed instructions.

Submitting the Assignment

This assignment is a combination of written and programming questions. Both portions of the assignment should be submitted through Gradescope.

The "Homework 6: Fun with Graphs" assignment is the written portion, for which you should submit a **typed** response to the non-coding questions (questions 1-1). Each response should clearly be marked with its corresponding number. You are free to use the provided templates, print the questions and write your answers, or to simply type your responses on a blank document (whatever works for you).

The "Homework 6: Coding" is the programming portion of the assignment. For this portion, download the ".zip" file from replit and upload this ".zip" file as your answer to Gradescope. You can upload the assignment as many times as you want.