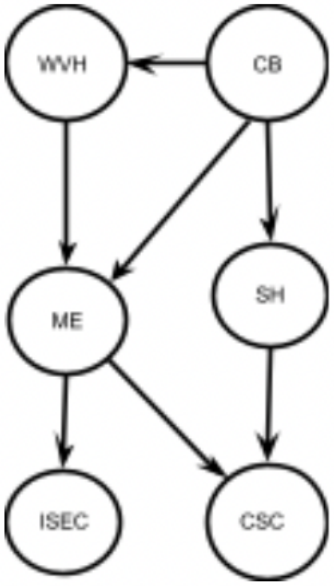
CS3000: Algorithms & Data — Summer 2023 — Laney Strange

Homework 3 - Short
Due Thursday June 8 @ 9pm Gradescope

Name: Laney Strange
Collaborators: the Banshees

- Put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Thursday June 8 @ 9pm Gradescope. You may submit up to 48 hours late for no penalty, but expect a delay in grading.
- Show ALL your work, even if the problem doesn't specify it.
- You'll have an opportunity to resubmit two homeworks (one long, one short) at the end of the semester.
- Solutions must be typeset, preferably in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- If you get stuck on a homework problem, come by office hours or post on Piazza! We recommend you spend about 30 minutes trying to figure out a problem, and then ask for help. We'll be happy to clarify material from class and algorithm concepts, but we will not give out solutions or confirm your answers are correct.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class, is strictly forbidden.

Problem 1. *Topological Sort (4 + 2 + 2 = 8 points)*



(a) Show the start and finish of vertices produced by DFS, if you start at CB, and the corresponding topological sort. Assume ties are broken in alphabetical order.

Solution:

(b) Another way to perform topological sort on a directed acyclic graph G is to repeatedly find a vertex of in-degree 0, output it, and remove it and all of its outgoing edges from the graph. Show the order of vertices this approach would produce.

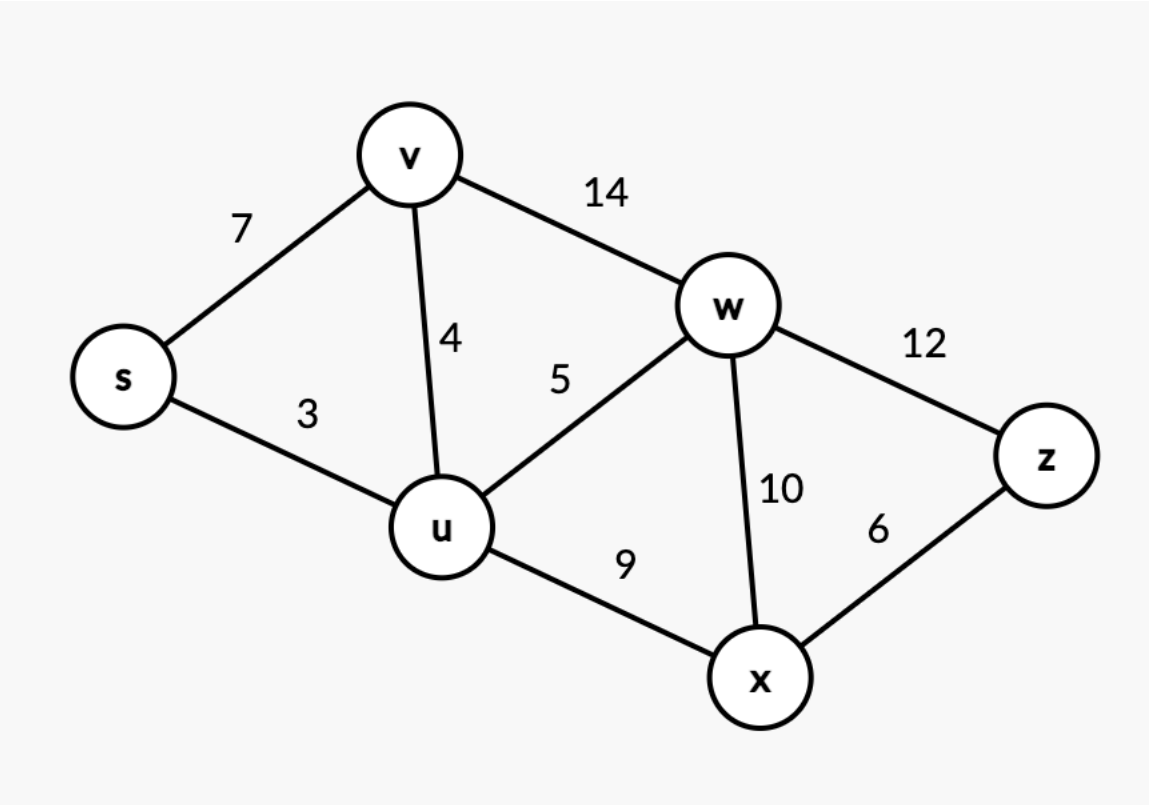
Solution:

(c) Briefly describe how you'd count the in-degree of all the vertices, assuming we use an adjacency-list representation for G. (You don't need to worry about the rest of the topological algorithm for this, just the initial degree-counting step.) What would the run-time of your approach be?

Solution:

Problem 2. *MSTs – Kruskal's (2 + 2 + 2 = 6 points)*

This problem is concerned with the graph below, and with Kruskal's MST algorithm, which you can find at https://course.ccs.neu.edu/cs3000/resources/Kruskal_Pseudocode.pdf



(a) If we follow Kruskal's algorithm, which edge would be considered first for an MST?

Solution:

(b) Which are the first five edges to be examined in Kruskal's? Would any be left out of the final set A?

Solution:

(c) What would the total weight of the MST be for this graph?

Solution:

Problem 3. *MSTs – Prims's (2 + 2 + 2 + 2 = 8 points)*

This problem is concerned with the same graph as for Problem 2. Now our focus is on Prim's algorithm, which you can find at https://course.ccs.neu.edu/cs3000/resources/Prim_Pseudocode.pdf

(a) Suppose Prim's algorithm picks s as its starting vertex. What are next two vertices considered?

Solution:

(b) Here is are the π and *key* values at the beginning of Prim's.

	s	u	v	w	x	z
key	0	∞	∞	∞	∞	∞
π	NIL	NIL	NIL	NIL	NIL	NIL

What will this table look like once the first vertex has been extracted and the loop from lines 10-14 has completed all its iterations (i.e., all the times the for loop runs on the first iteration of the while loop)?

Solution:

(c) What is in the heap H at the same point in the algorithm?

Solution:

(d) What will the table look like once Prim's has completed?

Solution: