CSCI 435: Algorithms and Complexity
Assignment 3
Due April 2, 2025 at 2:30pm

Assignment Regulations.

- This assignment must be completed individually.

- Please include your full name and email address on your submission.

- You may either handwrite or typeset your submission. If your submission is handwritten, please ensure that the handwriting is neat and legible.

[8 marks]   1. A student, who had a bit too much at the pub, is making their way home for the night. Fortunately, they're on the correct street, but they don't remember in which direction they need to walk in order to get home. The student can walk either east or west along the street, and they'll only know that they've arrived at home when they're standing outside of the house.

The student decides to use a "zig-zag" strategy to get home: starting from their current position (the origin), the student walks 1 unit east and checks if they're at home. If not, then the student walks back to the origin followed by another 2 units west. If the student didn't encounter their house in those 2 units, then they walk back to the origin followed by another 4 units east. If the student still didn't encounter their house in those 4 units, then this process continues: flipping the direction, returning to the origin, and doubling the additional distance each time.

Clearly, the optimal (offline) strategy would be if the student remembered that they lived d units away from the origin and simply walked d units in the appropriate direction.

Prove that the "zig-zag" strategy is a 9-competitive online algorithm for this house-finding problem; that is, if the student lives d units away from the origin, then they will travel a distance of at most $9d$ units to get home.

[5 marks]   2. Suppose we have a stack whose size never exceeds some constant k. We perform a sequence of PUSH and POP operations on this stack, but after every k operations, we back up the entire contents of the stack by copying them to some external memory location.

Show, using any appropriate method, that the cost of n stack operations (including copying) is $O(n)$. You may assume that the COPY operation is not included in the sequence of operations, but is instead called automatically after every k PUSH or POP operations.

[12 marks]   3. Consider the *shortest path* optimization problem:

> SHORTEST-PATH-OPT
> Given: a directed graph $G = (V, E)$, costs $c_e \geq 0$ for each $e \in E$, and two vertices $s, t \in V$
> Determine: a path P in G from s to t where $\sum_{e \in P} c_e$ is minimized

   (a) Write a linear program to solve the shortest path optimization problem.

   Hint 1. Use 0–1 decision variables to determine whether or not an edge is included in the path.

   Hint 2. Remember that any feasible path must start at s and end at t. Use constraints to ensure that you are guaranteed to follow some edge out of s and some edge into t, potentially with intermediate edges between the two vertices. You may assume that s has no incoming edges and t has no outgoing edges.

(b) Consider the following directed graph:
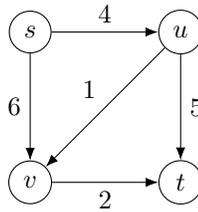
Using your template from part (a), write the linear program corresponding to the instance of the shortest path optimization problem involving this graph.

(c) Use your favourite mathematical software to solve your linear program from part (b) and find the shortest path in the graph.

There are many linear program solvers available online, in programming languages such as Python (via the SciPy package), and in software such as Maple, Mathematica, MATLAB (or its free equivalent, Octave), and even Microsoft Excel. Whatever approach you use, include your code or screenshots of the software producing the answer.

[5 marks] 4. Choose your favourite topic from the course, and write a multiple-choice style question with one correct answer and 3–4 plausible-but-incorrect answers that tests a concept or notion related to that topic.

For inspiration, consider the multiple-choice style questions you saw on the midterm and practice midterm exams.