St. Francis Xavier University
Department of Computer Science

CSCI 355: Algorithm Design and Analysis
Assignment 2
Due October 9, 2025 at 11:30am

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.

[6 marks] 1. (a) Consider a weighted graph $G = (V, E)$ where there exists some edge $e \in E$ having maximal weight and belonging to some cycle of edges in G. Prove that no minimum spanning tree of G will include the edge e.

(b) Consider a weighted graph $G = (V, E)$ where there exists some edge $e \in E$ having minimal weight. Prove that every minimum spanning tree of G will include the edge e.

[8 marks] 2. You have been hired as the equipment manager for the greatest team in the NHL: the Toronto Maple Leafs. As part of your job, you must assign hockey sticks to players.

Suppose you have a stock of n hockey sticks and the team has n players. Each hockey stick and each player has a size: for the sake of simplicity, suppose these sizes are numbered 1, 2, and so on. (Note that sizes need not be contiguous; for example, you may have a size 1 stick and a size 3 stick, but no size 2 stick.) Ideally, a size 1 player should use a size 1 stick, a size 2 player should use a size 2 stick, and so on. However, any player *could* use any size stick, though they'll complain about having to do so.

Your goal is to figure out how to distribute the n sticks to the n players in a way that minimizes the total difference in size between all players and their assigned sticks.

(a) Consider the following algorithm. Begin by assigning as many perfect matches as possible (i.e., matching size i sticks to size i players). Next, assign as many sticks to players as possible where the difference in size is 1. Then, assign as many sticks to players as possible where the difference in size is 2, and so on until all n players have sticks.

Although this algorithm sounds intuitive, it is not optimal. Prove this by giving a counterexample where the algorithm produces a larger total difference in size than the best possible.

(b) Design a greedy algorithm to solve this hockey stick matching problem. Prove that your algorithm runs in polynomial time, and argue using any appropriate method that your algorithm is optimal.

[6 marks] 3. The brand-new St. Francis Xavier Press has started operations, and employees must now begin to print books that will soon be published. However, the press has a limited number of printers. Every week, the press receives a list of books soon to be published, with each book i requiring time t_i to complete its print run. When a book i is scheduled for a print run, it also comes with a completion time C_i by which all books must be printed and sent to bookstores. Lastly, each book has a weight w_i that indicates how hotly anticipated it is: the higher the weight, the more readers want the book!

The press decides to order print runs in such a way that the weighted sum of the completion times, $\sum_{i=1}^{n} w_i C_i$, is minimized. Design an algorithm that takes a set of n books, each with a printing time t_i and weight w_i, and produces a schedule that minimizes this weighted sum of completion times. Prove that your algorithm produces an optimal solution.

[5 marks] 4. As Canadian drivers well know, weather has a big impact on the time required to drive from one place to another. Travelling on a clear, dry road takes much less time than travelling on a wet, snowy road. For this reason, a group of enterprising computer science students has created an app that predicts how much time will be needed to drive between two locations based on the current season. s This app handles input in the form of a road $r = (a, b)$ between two locations a and b together with a departure time t, and returns a value $f_r(t)$ indicating the predicted travel time between departing a at time t and arriving at b. The value $f_r(t)$ is arbitrary, aside from two conditions: $f_r(t) \geq t$ for all roads r and times t (you can't travel back in time) and $f_r(t)$ is a monotone increasing function (you can't arrive earlier by departing later).

Construct an algorithm that uses these values $f_r(t)$ to find the fastest route between any two locations a and b. You do not need to prove the correctness of your algorithm, but it should run in polynomial time.

Hint. Can we modify an existing graph algorithm to work for this problem?