St. Francis Xavier University
Department of Computer Science

CSCI 355: Algorithm Design and Analysis
Assignment 1
Due February 10, 2022 at 1:15pm

Assignment Regulations.

- This assignment may be completed individually or in a group of up to four people. If you are collaborating on an assignment as a group, your group must submit exactly one joint set of answers.
- Please include your full name and email address on your submission. For groups, every member must include their full name and email address on the joint 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. Shinobi and Pokanys, two streamers on the popular video game streaming website Twinge, are planning their schedules for the new year. There are n streaming slots, and each streamer has n video games that they enjoy streaming.

Each video game has an integer rating, calculated from the number of viewers for that video game in the past year. We can assume no two video games have the same rating. A streamer wins a given streaming slot if the video game they chose for that slot has a higher rating than the game chosen by the other streamer for the same slot. The goal of each streamer is to create a schedule of video games to streaming slots $S = \{g_1, g_2, \ldots, g_n\}$ that maximizes the amount of slots they win.

On the first day of the new year, Shinobi posts his schedule S and Pokanys posts her schedule T. This pair of schedules determines which streamer wins which slots. We say that a pair of schedules (S, T) is *stable* if there exists no schedule S' where Shinobi wins more slots with the pair (S', T) and, likewise, there exists no schedule T' where Pokanys wins more slots with the pair (S, T').

Unfortunately, unlike the Gale–Shapley algorithm, we cannot guarantee the existence of a stable pair of schedules in this scenario. To demonstrate this, give an example of a set of video games and their associated ratings for both Shinobi and Pokanys where there exists no stable pair of schedules.

[6 marks] 2. Arrange the following functions in order from slowest growth rate to fastest growth rate. If function $f(n)$ is in position i of your list and function $g(n)$ is in position $i+1$ of your list, then this is equivalent to saying that $f(n) \in O(g(n))$. Give a brief justification for your ordering.

$$\begin{array}{ll} f_1 = n^{3.5} & f_4 = 2^{n^2} \\ f_2 = \sqrt{16n} & f_5 = 50^n \\ f_3 = 42 + n & f_6 = n^3 \log(n) \end{array}$$

[5 marks] 3. Consider the following problem: given a one-dimensional array A containing n integers, compute a two-dimensional array B where entry $B[i,j]$, $i < j$, stores the sum of array entries $A[i] + A[i+1] + \cdots + A[j]$. You can assume that all entries $B[i,j]$ with $i \geq j$ are zero.

The following algorithm solves this problem:

Algorithm: Subarray addition

> **for** $1 \leq i \leq n$ **do**
>   **for** $(i+1) \leq j \leq n$ **do**
> $B[i,j] \leftarrow$ sum of array entries $A[i]$ through $A[j]$

(a) Establish a bound $O(f(n))$ on the running time of this algorithm, where $f(n)$ is some function and n is the size of the input given to the algorithm.

(b) Using your function $f(n)$ from part (a), establish a matching bound $\Omega(f(n))$ on the running time of this algorithm. Again, n is the size of the input given to the algorithm.

Hint. Consider the case when $i \leq \frac{n}{4}$ and $j \geq \frac{3n}{4}$. What is the least number of operations required to perform the summation within the inner for loop?

(c) What do your two bounds from parts (a) and (b) imply about the performance of this algorithm?

[8 marks] 4. A group of ornithology researchers has asked you to help them track the spread of a novel crow virus called Corvid. The researchers have tagged n birds in their system and named them B_1 through B_n. They give you a set of m data tuples indicating when pairs of birds were detected together: this data is of the form (B_i, B_j, t_k), indicating that birds B_i and B_j were together at time t_k.

If one infected bird B_i is detected with an uninfected bird B_j at time t_k, then bird B_j becomes infected from time t_k onward. This infection is modelled by the existence of either of the tuples (B_i, B_j, t_k) or (B_j, B_i, t_k). The spread of the virus can then be modelled by a sequence of tuples: if bird B_i is infected by time t_k, and there are tuples (B_i, B_j, t_k) and (B_j, B_m, t_ℓ) where $t_k \leq t_\ell$, then bird B_m will be infected via bird B_j.

Design an algorithm that answers the following question: if bird B_a was infected by the virus at time t_x, could it have infected bird B_b by time t_y? Your input is the set of m tuples defined earlier, as well as B_a, B_b, t_x, and t_y. You can assume that the tuples are sorted by time, that each pair of birds is found together at most once, and that a bird remains forever infectious once it is infected.

You do not need to establish the correctness or running time of your algorithm, but it may help to know that the official solution runs in linear time.