St. Francis Xavier University
Department of Computer Science

CSCI 355: Algorithm Design and Analysis
Assignment 1
Due January 26, 2023 at 1:15pm

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. In our discussion of the stable matching problem, we assumed that our preference lists were totally ordered: in a given list, the first option was preferred over the second option, the second over the third, and so on. Here, we will consider a variant of the problem where we allow for indifference between options.

Suppose we have a set S of n students and a set H of n hospitals, with their associated preference lists, but now either list can indicate a tie between two students or two hospitals. As an example, when $n = 4$, one hospital could rank student s_1 first, rank both students s_2 and s_3 second (meaning that the hospital has no preference between these two students), and rank student s_4 last. As before, we say that hospital h *prefers* student s over student s' if s is ranked strictly higher than s' on the preference list of h (i.e., s and s' are not tied).

(a) Say that a perfect matching M has *strong instability* if there exists a student s and a hospital h such that both s and h prefer the other to their assigned match in M.

Does there always exist a perfect matching with no strong instability? If so, give a procedure that is guaranteed to find a perfect matching with no strong instability. If not, give an example of a set of students S and a set of hospitals H whose preference lists are such that every perfect matching has a strong instability.

(b) Say that a perfect matching M has *weak instability* if there exists a student s and a hospital h such that (i) their matches are h' and s', respectively, and (ii) one of the following properties holds:

- s prefers h over h', and h prefers s over s' or is indifferent; or
- h prefers s over s', and s prefers h over h' or is indifferent.

Does there always exist a perfect matching with no weak instability? If so, give a procedure that is guaranteed to find a perfect matching with no weak instability. If not, give an example of a set of students S and a set of hospitals H whose preference lists are such that every perfect matching has a weak instability.

[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.

$$
\begin{array}{ll}
f_1 = 4^{\log(n)} & f_4 = (3/2)^{n+1} \\
f_2 = \mathrm{e}^n & f_5 = n + n\log(n) \\
f_3 = \ln(n) & f_6 = n \cdot 2^n
\end{array}
$$

[4 marks] 3. For each of the following blocks of pseudocode, give an analysis of the time complexity with as tight a bound as possible. You do not need to give a formal proof, but you should give both upper and lower bounds together with justifications for each of your answers.

(a)
```
for i from 1 to n*n:
    j = n
    while j > 0:
        j = j-1
```

(b)
```
x = 0
for i from 1 to n*n:
    for j from 1 to i*i:
        x = x+1
```

[7 marks] 4. You have just gotten a job at a Canadian telecommunications company that has been in the news for network outages. They're getting tired of the bad press, so they want you to prove that if a node that is *just* far enough away from their headquarters goes down, they can't be blamed for the network outage.

The problem can be formulated as follows. Suppose the network contains n nodes, and consider the company's headquarters at node s together with some other specified node t. Assume that the distance between s and t is strictly greater than $n/2$. You must prove to the boss of the company that there exists some node v between s and t (where $v \neq s$ and $v \neq t$) such that, if v is removed from the network, then all s–t paths are severed.

Hint. Show that there must exist at least one "layer" between s and t that consists of a single node.