Instructions. This assignment should be completed individually with no collaboration at all.

Problem 1. This problem is designed to refresh your analytical skills.

Rank the following function by order of growth. Functions of different orders should be listed on separate lines in order of their growth rates, fastest growth rates at the top of page. If you have two functions \( f(n) \) and \( g(n) \) in the same class, make sure to list them on the same line.

\[
\begin{array}{c|c|c|c|c|c}
\text{Function} & n^2 & \log_2 n & n! & n^3 & n^2 \log(n) + 1
\end{array}
\]

Problem 2. This problem is for you to practice writing pseudocode. Expect more problems of this type throughout the semester.

Consider the searching problem:

Input: A list of \( n \) numbers \( A = < a_1, a_2, ..., a_n > \) and a value \( v \).

Output: An index \( i \) such that \( v = A[i] \), or the special value NIL if \( v \) does not appear in \( A \).

1. Write pseudocode for the linear search, which scans through the sequence, looking for \( v \). Hint: You might need a loop!

2. What is the Big-Oh performance of the algorithm you defined?

3. In your own words, show that your algorithm is correct. Remember, your explanation should be correct, clear, and concise.

Problem 3. This problem is for you to go back to notes from previous courses, practice writing pseudocode, and perform simple algorithm analysis. Expect more of this in the semester.

Remember the Insertion sort algorithm? It’s a simple sorting algorithm that sorts arrays one element at a time. Answer the following questions for the implementation shown below.

\[
\text{InsertionSort}(A)
\]

\[
\begin{align*}
\text{i} &= 2 \\
\text{while i <= length(A)} \\
\text{x} &= A[i] \\
\text{j} &= i - 1 \\
\text{while j > 0 and A[j] > x} \\
\text{A[j+1]} &= A[j] \\
\text{j} &= j - 1 \\
\text{end while} \\
\text{A[j+1]} &= x \\
\text{i} &= i + 1 \\
\text{end while}
\end{align*}
\]

a) Analyze the worst case running time complexity of the algorithm. Show your work.

Problem 4. In page 8 of the textbook, you will find the proof for the correctness of the Stable Matching problem. Please read it, and rewrite it in your own words.