Problem 1. Consider the searching problem:

INPUT: A list of \( n \) numbers \( A = \langle a_1, a_2, \ldots, a_n \rangle \) and a value \( v \).

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

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

(b) What is the Big-O performance of the algorithm you defined?

(c) Argue why your algorithm is correct. Try to write a clear and concise explanation.

Problem 2. The Insertion sort algorithm is a simple sorting algorithm that sorts arrays one element at a time. Analyze the worst case running time complexity of the implementation below. Show your work.

```plaintext
InsertionSort(A)
   i = 2
   while i <= length(A)
      x = A[i]
      j = i - 1
      while j > 0 and A[j] > x
         j = j - 1
      end while
      A[j+1] = x
      i = i + 1
   end while
```

Problem 3. Decide whether you think the following statements are true or false. If it is true, give a short explanation. If it is false, give a counterexample.

(a) In every instance of the Stable Matching Problem, there is a stable matching containing a pair \((m, w)\) such that \( m \) is ranked first on the preference list of \( w \) and \( w \) is ranked first on the preference list of \( m \).

(b) Consider an instance of the Stable Matching Problem in which there exists a man \( m \) and a woman \( w \) such that \( m \) is ranked first on the preference list of \( w \) and \( w \) is ranked first on the preference list of \( m \). Then in every stable matching \( S \) for this instance, the pair \((m, w)\) belongs to \( S \).
Proof Problem

\LaTeX\ source for template: https://www.overleaf.com/read/wrgwwcsrmvcq

Here we present a different version of the problem of matching residents and hospitals. In this scenario, we have $m$ hospitals, each with a certain number of available positions for hiring residents. We have $n$ medical students graduating in a given year, each interested in joining one of the hospitals. Each hospital has a ranking of the students in order of preference, and each student has a ranking of the hospitals in order of preference. We assume that there are more students graduating than there are slots available in the $m$ hospitals.

The goal is to find a way of assigning each student to at most one hospital, in such a way that all available positions in all hospitals are filled. (Since we are assuming a surplus of students, there are some students who do not get assigned to any hospital.)

We say that an assignment of students to hospitals is stable if neither of the following situations arises.

• First type of instability: There are students $s$ and $s'$, and a hospital $h$, so that
  
  – $s$ is assigned to $h$, and
  – $s'$ is assigned to no hospital, and
  – $h$ prefers $s'$ to $s$.

• Second type of instability: There are students $s$ and $s'$, and hospitals $h$ and $h'$, so that
  
  – $s$ is assigned to $h$, and
  – $s'$ is assigned to $h'$, and
  – $h$ prefers $s'$ to $s$, and
  – $s'$ prefers $h$ to $h'$.

So we basically have the Stable Matching Problem, except that (i) hospitals generally want more than one resident, and (ii) there is a surplus of medical students.

(a) Show that there is always a stable assignment of students to hospitals, considering the two types of instability defined above.

(b) Give an algorithm that finds a stable assignment.