MAT2440/D641 **Quiz #4**
Instructor: Suman Ganguli **Monday, November 19** **Name:** _____

Question:	1	2	3	Total
Points:	3	4	3	10
Score:				

Here is pseudocode for the linear search algorithm:

procedure linear-search (x : integer, $a_1, a_1, \ldots a_n$: distinct integers)
  $i := 1$
  while $(i \leq n)$ and $x \neq a_i$:
    $i := i + 1$
  if $(i \leq n)$ **then**:
    location := i
  else:
    location := -1
  return location

1. (3 points) Briefly explain (in 3-4 sentences) what the linear search algorithm does and how it works.

   (Some questions to consider in your explanation: What is the algorithm supposed to accomplish? What does the algorithm take as inputs, and what does it return as output? What does the output represent? What are the steps the algorithm takes to get from the inputs to the outputs?)

   Solution:

   The algorithm searches for the input x in the input list $a_1, \ldots a_n$, and it outputs the location of x if x is found in the list; if x is not found in the list, the algorithm returns -1. The algorithm does this by starting with $i = 1$, and checking if $a_1 = x$. If so, it returns location $= 1$; if not, it increments i to $i = 2$. It then checks if $a_2 = x$. In this way, it steps thru the entire list, comparing each element a_i with x, until it finds $a_i = x$. If such an a_i is found, it returns that i as location. If x is not in the list, then i gets incremented to $n + 1$, and the algorithm returns -1.

2. (4 points) Now let's consider how long the linear search algorithm takes to terminate. How long the algorithm takes to terminate will vary, depending on the inputs. In particular, the number of times the "while" loop is executed determines how long the the algorithm takes.

   a. Suppose $a_1 = 3, a_2 = 8, a_3 = 9, a_4 = 15$. Which will take longer to run:

      (i) "linear-search $(x = 15 : a_1, a_2, \ldots a_n)$" or

      (ii) "linear-search $(x = 3 : a_1, a_2, \ldots a_n)$"?

      In particular, how many times will the "while" loop be executed in (i), vs. how many times will it be executed in (ii)?

      Solution:

      (i) will take longer, since the algorithm searches thru the entire list before finding x in the last position, whereas for (ii) the algorithm terminates right away, since it finds x in the first position.

      In (ii), the "while" loop will be executed 0 times (the algorithm never actually enters the while loop, since $x = a_1$, and jumps right away to the "if" statement to return location = 1).

      In (i), the "while" loop will be executed 3 times: to increment i from 0 to 1, from 1 to 2, and from 3 to 4. When $i = 4$, the algorithm that $x = a_4$, and then escapes the loop to the "if" statement.

b. Given a list $a_1, a_2, \ldots a_n$ of length n, for what value of x does "linear-search $(x : a_1, a_2, \ldots a_n)$" take the longest to run? How many steps (i.e., how many "while" loop executions) does it take in that case? (This is called the "worst-case" performance of the algorithm; we will discuss this in Section 3.3.)

Solution:

The worst-case scenario is if x is not in the list at all, i.e., $x \neq a_1, x \neq a_2, x \neq a_n$. Then linear search compares x with each a_i in succession, incrementing i in each execution of the while loop, until $i = n+1$. It thus executes the while loop n times in this case.

3. (3 points) Find the terms a_2, a_3, a_4 in the sequence defined by the following recurrence relation and initial conditions:

$$a_n = a_{n-1} + 3a_{n-2}$$
$$a_0 = 1, a_1 = 2$$

Solution:

- $a_2 = a_1 + 3a_0 = 2 + 3(1) = 5$
- $a_3 = a_2 + 3a_1 = 5 + 3(2) = 5 + 6 = 11$
- $a_4 = a_3 + 3a_2 = 11 + 3(5) = 11 + 15 = 26$