CSC 321 Algorithm Analysis and Design
Homework Assignment 1

Due on 02/7/2021 11:59PM

Problems

1. **Comparison of running times**
   For each function $f(n)$ and time t in the following table, determine the largest size n of a problem that can be solved in time t , assuming that the algorithm to solve the problem takes $f(n)$ microseconds.

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\lg n$							
$\sqrt{n}$							
n							
$n \lg n$							
n^2							
n^3							
2^n							
$n!$							

2. **Nonincreasing Insertion Sort**
   Rewrite the INSERTION-SORT procedure to sort into nonincreasing instead of nondecreasing order.

3. **Linear Searching**
   Consider the ***searching problem***:

   Input: A sequence 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. Write pseudocode for ***linear search***, which scans through the sequence, looking for v. Using a loop invariant, prove that your algorithm is correct. Make sure that your loop invariant fulfills the three necessary properties.

4. **Adding Binary Numbers**
   Consider the problem of adding two n-bit binary integers, stored in two n-element arrays A and B. The sum of the two integers should be stored in binary form in an $(n+1)$-element array C. State the problem formally and write pseudocode for adding the two integers.

5. **Recursive Insertion Sort**
   We can express insertion sort as a recursive procedure as follows. In order to sort $A[1..n]$, we recursively sort $A[1..n-1]$ and then insert $A[n]$ into the sorted array $A[1..n-1]$. Write a recurrence for the running time of this recursive version of insertion sort.