COMP 2119B Assignment 1

Instruction: Submit your solutions of problems in Part I only. Questions in Part II are ungraded. No need to submit your work for Part II. Due Date: 04 February.

Part I

Question 1. Given the following recurrence equation:

$$T(n) = \begin{cases} 1 & n = 1 \\ 27T(\frac{n}{3}) + n^2 & n > 1, \end{cases} \tag{1}$$

where $n = 3^k$. Express $T(n)$ in terms of n only.

Question 2. Read the following pseudocode.
(a): Define a recurrence equation for $G(n)$.

```
1: function G(n)
2:     if n=0 then return 0
3:     else if n=1 then return 1
4:     else if n=2 then return 2
5:     else
6:         return G(n − 1) + G(n − 2) + G(n − 3)
7:     end if
8: end function
```

(b): What is the number of times that addition '+' operations in line 6 are executed?

Question 3. Use mathematical induction to prove that:
(a): for any integer $n > 23$, there exist non-negative integers x and y such that $n = 7x + 5y$;
(b): the nth Fibonacci number, $f(n)$, has the following closed form:

$$f(n) = f(n-1) + f(n-2) = \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n],$$

with $f(0) = 0$ and $f(1) = 1$.

Question 4. Deign a recursive algorithm for the following problems and prove your claim:
(a): Given a sorted array $A[1, ..., n]$ of distinctive non-negative integers in a decreasing order, determine whether there is an index i with $A[i] = i$;
(b): Given an unsorted array $A[1, ..., n]$, find out the maximum value $A[j] - A[i]$ over all $1 \le i, j \le n$.

Part II

Q1 Let $f(n) = n^3 + n$. Find the error in the following M.I. proof that $f(n) \in O(n^2)$.

1) Basis: For $n = 1$, we have $f(1) = 1 + 1 = 2 = cn^2$, where $c = 2$. So the case is trivially satisfied.

2) Induction Step: For any $n > 1$, assume by the induction hypothesis that there exists a constant $c > 0$ such that $f(n-1) \le c(n-1)^2$. Then,

$$f(n) = n^3 + n = (n-1)^3 + (n-1) + 3n^2 - 3n + 2 = f(n-1) + 3n^2 - 3n + 2$$

$$\le c(n-1)^2 + 3n^2 - 3n + 2 = cn^2 - 2cn + c + 3n^2 - 3n + 2$$

Take c such that $c \ge \frac{3n^2-3n+2}{2n-1}$, then $f(n) \le cn^2$ for some constant c.

So, we showed that no matter how large n is, we can always find a value of c such that $0 \le f(n) \le cn^2$. It follows that $f(n) \in O(n^2)$.

Q2 a) Write a recursive algorithm, $power_3(n)$ that computes 3^n for some positive integers n based on the following equality:

$$3^n = 3^{n-1} + 3^{n-1} + 3^{n-1}$$

b) Let $f(n)$ be the number of additions executed by $power_3(n)$, formulate a recurrence equation of $f(n)$
c) Solve $f(n)$.