CSC 355. Discrete Structures and Basic Algorithms
Homework Assignment 2

Instructions: Solve the following questions.

1. Using induction to prove that for any natural number *n, 0+1+…+ n = n(n+1)/2*
2. Using the literature provided in Module 2, explain the following:
    a. Describe the upper bound.
    b. Describe the lower bound.
    c. Describe the grow rate.
3. State the recursive algorithm or pseudocode to solve the following exercises. For each algorithm analyze the best, worst and average cases.
    a. Fibonacci Series
    b. Factorial
    c. Hanoi Tower
4. Does the Linear Search is Ω(1) in its best case? Why?
5. The sequential search algorithm is $\Theta(n^2)$ or not? Why?

Submission Instructions

You must upload your homework in a **pdf** file in the designated area in D2L.

Grading Points

Total Score: 25 points

**Each question has a value of 5 points*

(1) Assume $0+1+2,\ldots,n = n(n+1)/2$ is true for $n=k$ where $k=5$

$$0+1+2+3+4+5 = 5(6)/2 = 15$$

if $0+1+2,\ldots,+(k+1) = (k+1)(k+2)/2$ then $0+1+2,\ldots,n = n(n+1)/2$ is true for all n

proof: $k+1=6$

$$0+1+2+3+4+5+6 = 21$$

$$6(7)/2 = 42/2 = 21$$

■

(2)

a. Worse case scenerio for an algorithm in runtime

b. Best case scenerio for an algorithm in runtime

c. The average scenerio for an algorithm in runtime

(3)

a.

```
Fibonacci(n){
        if (n == 0)
                return 0;
        else if (n == 1)
                return 1;
        else
                return Fibonacci(n-1) + Fibonacci(n-2)
}

// Best case is n = 0; T = 1
// Worse case is n is very large; O(2^n)
// Average case is any number n; O(2^n)
```

b.

```
Factorial(n) {
        if (n == 0)
                return 1;
        return n * Factorial(n - 1)
}

// Best case is n = 0; T = 1
// Worst case is N is very large; O(N)
// Average case is N is a number; O(N)
```

c.

```
Hanoi(n, src, aux, tgt){
        if (n == 1)
                move disk 1: src -> tgt
        else
                Hanoi(n - 1, src, tgt, aux)
                move disk n: src -> tgt
                Hanoi(n - 1, aux, src, tgt)
}

// Best case is n = 1; T = 1
// Worst case case is N is very large; O(2^N)
// Average case is N is a number; O(2^N)
```

(4.) Yes, because in the best case scenario it would only take one iteration to find the matching item.

(5.) No, because the worst case scenario (big O) of a sequential search is n. This means that the theta value can't be larger than big O.