CSCI 4452.2 2014 Assignment #3

Show all your work.

(1) Consider the following recursive implementation of the Fibonacci function

```
int Fibonacci(N)
    if N == 1 or N == 2
        return 1
    else
        return (Fibonacci(N-1) + Fibonacci(N-2))
    end if
end Fibonacci
```

Write down the corresponding cost function to count the number of additions and subtractions (count total number of additions and subtractions together) for the input N (explain how this cost function arises from the Fibonacci function) and then perform an analysis of this cost function *using the characteristic polynomial approach.* Also, how many multiplications does this algorithm perform? How many additions and subtractions does it perform for the input $N = 10$? How many multiplications for $N = 10$?

(2) Develop a non-recursive implementation of the Fibonacci function that uses an iterative, loop-based approach. Write down the corresponding cost function (to count the number of additions and subtractions) for the input N. How many additions and subtractions does this implementation of the function perform? Also, how many multiplications does this algorithm perform? How many additions and subtractions does it perform for the input $N = 10$? How many multiplications for $N = 10$?

(3) Develop a nonrecursive, non-iterative, implementation of the Fibonacci function that uses the closed form expression for the evaluation of the Nth element of the Fibonacci sequence developed in class. Analyze this function to determine the number of additions and subtractions this algorithm performs. Also determine the number of multiplications this algorithm performs. How many additions and subtractions does it perform for the input $N = 10$? How many multiplications for $N = 10$?

(4) Write a program (in the language of your choice) that implements the Fibonacci algorithm from Question (1). Add counters to keep track of the number of additions and subtractions and the number of multiplications that are performed.

(5) Repeat Question (4) for the iterative implementation of the Fibonacci function you developed in Question (2).

(6) Repeat Question (4) for the non-recursive, non-iterative implementation of the Fibonacci function you developed in Question (3).

(7) How well do the experimental counts compare with those from the algorithm analyses? How robust are the implementations? That is, do the functions compute the correct answers? What is the largest value of N that can be handled by each implementation? Discuss your results.