COMP 3761 Assignment 2

Due: Wednesday **Jan 28, 2015** at 6:30pm

1. Consider the following algorithms:

   (A)
   ```
   long A(int n) {
       if (n == 1)
           return 3;
       return 2 * A(n - 1) + 3;
       }
   ```

   (B)
   ```
   long B(int n) {
       if (n == 1)
           return 1;
       return B(n / 2) + n;
   }
   ```

   (C)
   ```
   long C(int n) {
       if (n==0)
           return 0;
       return C(n - 1) + n;
   }
   ```

   (D)
   ```
   long D(int n) {
       if (n==1)
           return 1;
       return D(n / 3) + 1;
   }
   ```

   Assume that the input value n is a positive integer.

   (a) Set up a recurrence relation for each function's values, including its initial condition. [1 point x 4]

   (b) Theoretically analyse the time efficiency for each recursive algorithm:

      i. Which operation is considered as the basic operation for the time efficiency analysis? [0.5 point x 4]

      ii. Set up a recurrence relation for the number of basic operation count, including its initial condition. [1 point x 4]

      iii. Solve the recurrence relation in (ii). Show the number of basic operations as an exact function of input size n. [1 point x 4]
      For your convenience, you may assume $n = 2^k$ for (B) and $n = 3^k$ for (D).

      iv. What is the worst-case time efficiency class of each algorithm? [0.5 point x 4]

2. Consider the following recursive algorithm:

```
void Secret(A[0..last]) {
    if (last == 0)
        return;
    Secret(A[0..last -1]);
    temp = A[last];
    int pos = last -1;
    while ((pos >= 0) && (A[pos] > temp)) {
        A[pos +1] = A[pos];
        pos --;
    }
    A[pos+1] = temp;
}
```

Given an array $A[0..n-1]$ of $n > 0$ orderable elements, initially $last = n-1$ in the above algorithm.

(a) What does this algorithm compute? [1 point]

(b) What operation is considered as its basic operation? [1 point]

(c) Set up a recurrence relation for the number of times the basic operation is executed in both the best and worst cases, for the input size n. [2 points]

(d) Mathematically solve the recurrence relations for the basic operation count in both the best and the worst cases. Express each count as an exact formula of n. Show your mathematical intermediate steps to support your answer. [2 points]

(e) What is the best-case and worst-case time efficiency classes of the algorithm? [2 points]

(f) Implement the algorithm in Java, assuming the array A contains integers only. [2 points]

(g) Given an array with the same known input size n, what kind of input data is considered as the best-case, worst-case, and average-case input for this algorithm? [3 points]

(h) Write a program to test the correctness and time efficiency of your implementation. Your program should read the input size n from the standard input (keyboard) and automatically populate the different types of input data arrays to test your method implementation. Run your experiments with different input sizes n and different types of input data, display your sample test results and comment on your test observations. [3 points]