Student Name: ______________________________ J#: ________________

CSC 323 Algorithm Design and Analysis, Spring 2016

Instructor: Dr. Natarajan Meghanathan

Quiz 3 (February 16, 2016) Max. Points: 25 Max. Time: 15 min.

1) (6 pts) Consider a text comprising of 500 ones. Determine the number of comparisons encountered to search for the following patterns in this text. Show all the work.

11111111111111111111 ... 111 (500 ones)

(a) 11101 (b) 01111

(a) 11101

It would take 4 comparisons per block to decide the pattern does not match for the block.

blocks = (500 − 5 + 1) = 496.

total # comparisons = 496 × 4 = 1984.

(b) 01111

It would take just 1 comparison per block to decide the pattern does not match for the block.

~~tot~~ # blocks = (500 − 5 + 1) = 496.

total # comparisons = 496 × 1 = 496.

2) (7 pts) Use a $\Theta(n)$ algorithm to determine the number of sub strings that start with a 'C' and end with a 'D' in the string ACDABCCADACD. Show all the work.

	A	C	D	A	B	C	C	A	D	A	C	D
# Cs	0	1	1	1	1	2	3	3	3	3	4	4
# substrings C...D	0	0	1	1	1	1	1	1	4	4	4	8

Student Name: ______________________________ J#: ________________

3) (12 pts) Consider the following pseudo code for insertion sort. Derive the expressions for the best-case and worst-case number of comparisons and the overall time-complexity of the algorithm with respect to an appropriate asymptotic notation.

```
ALGORITHM  InsertionSort(A[0..n − 1])
    //Sorts a given array by insertion sort
    //Input: An array A[0..n − 1] of n orderable elements
    //Output: Array A[0..n − 1] sorted in nondecreasing order
    for i ← 1 to n − 1 do
        v ← A[i]
        j ← i − 1
        while j ≥ 0 and A[j] > v do
            A[j + 1] ← A[j]
            j ← j − 1
        A[j + 1] ← v
```

Best-case $\sum_{i=1}^{n-1} 1 = (n-1-1+1) = n-1$

Worst-case $\sum_{i=1}^{n-1} \sum_{j=i-1}^{0} 1 = \sum_{i=1}^{n-1} \sum_{j=0}^{i-1} 1$

$= \sum_{i=1}^{n-1} [i-1-0+1] = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$

Overall $\lim_{n\to\infty} \frac{\text{best-case}}{\text{worst-case}} = \lim_{n\to\infty} \frac{n-1}{n(n-1)/2} = \lim_{n\to\infty} \frac{2}{n} = 0.$

Overall time complexity $= O\left(\frac{n(n-1)}{2}\right) = O(n^2)$