Student Name: ______________________________ J#: ________________

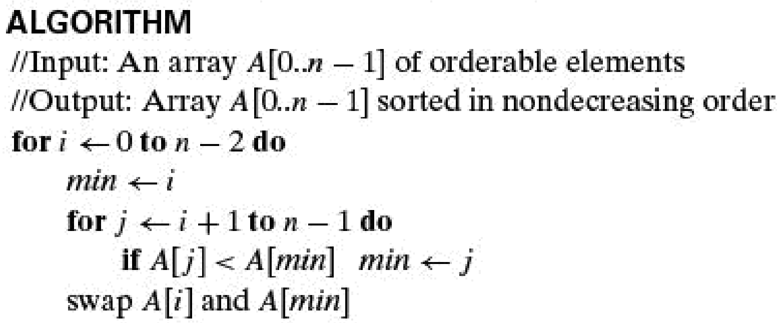
CSC 323 Algorithm Design and Analysis, Fall 2020

Instructor: Dr. Natarajan Meghanathan

Exam 1 (Take Home; Due by: September 24, 2020; 11.59 PM) Max. Points: 100

<u>Answering and Submission:</u> You should answer in the space provided. Either type your solutions or scan your answers and include the scanned picture file in the space provided in the document and submit (upload in Canvas) the entire exam as a single word document or PDF file.

Q1) (10 pts) Given the following pseudo code of an algorithm to sort the integers in ascending order:
(i) Identify the basic operation
(ii) Would there be a best and worst case or not? Justify your answer.
(iii) According to your answer for (ii), determine the number of times, the basic operation will be executed. Show all the work.



```
ALGORITHM
//Input: An array $A[0..n-1]$ of orderable elements
//Output: Array $A[0..n-1]$ sorted in nondecreasing order
for $i \leftarrow 0$ to $n-2$ do
    $min \leftarrow i$
    for $j \leftarrow i+1$ to $n-1$ do
        if $A[j] < A[min]$   $min \leftarrow j$
    swap $A[i]$ and $A[min]$
```

Q2) (7 pts) Derive the asymptotic relationship between the two functions: $n^{100}\log(n)$ and $n\log(n^{100})$

Q3) (7 pts) Let f(n) = $5n^3$ + 6n + 2. Find a function g(n) such that f(n) = O(g(n)) and f(n) ≠ Θ(g(n)). Show that your choice for g(n) is correct using the Limits approach.

Q4) (15 pts) The number of inversions is an array is the number of (i, j) pairs (where i and j are index positions and each pair is considered only once) such that A[i] > A[j] and i < j.

For example, the following array has 5 inversions as shown.

Index	0	1	2	3	4
Array, A	**10**	**50**	**40**	**20**	**30**

Index Pairs	Inversions
(1, 2)	A[1] > A[2]
(1, 3)	A[1] > A[3]
(2, 3)	A[2] > A[3]
(1, 4)	A[1] > A[4]
(2, 4)	A[2] > A[4]

(a) Modify the pseudo code of the Insertion Sort algorithm so that it can compute the number of inversions in an array. Write the modified pseudo code and justify your modification.
(b) Analyze whether the modification would have any impact on the asymptotic time complexity of Insertion sort.
(c) Run your modified version of the Insertion sort algorithm on the above array and show that it can determine the number of inversions to be 5.

Q5) (7 pts) Given the code snippet below, identify the basic operation and determine the number of times the basic operation is executed as a function of the input *n*.

```
for (int i = 1; i ≤ n; i++) {
     for (int j = n; j ≥ 1; j = j / 2){
          for (int k = 1; k ≤ n; k++){
              print "Hello World";
          }
      }
}
```

Q6) (5 pts) Let $T(n) = 7T(n/2) + 3n^2 + 2$.
Using Master Theorem and the Limits approach, show that $T(n) = O(n^3)$

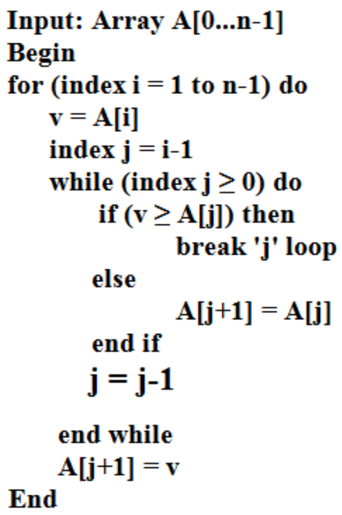
Q7) (5 pts) Let $T(n) = 3T(n/4) + n\log n$. Solve the recurrence using Master Theorem.

Q8) (14 pts) Solve the recurrence relation "without" using Master Theorem
T(n) = T(n/2) + 1, for n > 1
T(1) = 1

You will get a ZERO if you use Master Theorem to solve the recurrence relation.

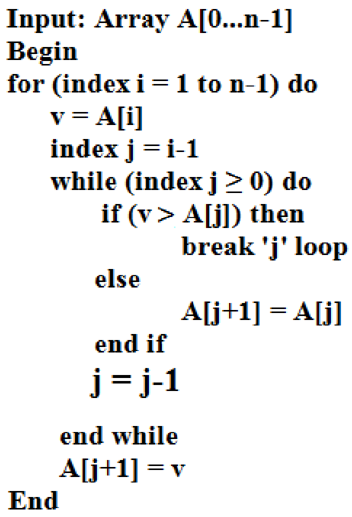
Student Name: ______________________________ J#: ________________

Q9) (15 pts) Consider the following two variants of the pseudo code for the Insertion Sort algorithm. Using each variant, sort the array: 5_1 5_2 5_3 5_4 5_5. Note that 5_1, 5_2, ..., 5_5 are five different instants of integer 5 and need to be treated as separate elements (that are of the same numerical value). Determine the number of comparisons encountered with each of the two variants of the algorithm to sort the above array and what is the final sorted array (include the suffixes of the elements throughout your work).



```
Input: Array A[0...n-1]
Begin
for (index i = 1 to n-1) do
   v = A[i]
   index j = i-1
   while (index j ≥ 0) do
      if (v ≥ A[j]) then
           break 'j' loop
      else
           A[j+1] = A[j]
      end if
      j = j-1
    end while
    A[j+1] = v
End
```

Pseudo Code - I



```
Input: Array A[0...n-1]
Begin
for (index i = 1 to n-1) do
   v = A[i]
   index j = i-1
   while (index j ≥ 0) do
      if (v > A[j]) then
           break 'j' loop
      else
           A[j+1] = A[j]
      end if
      j = j-1
    end while
    A[j+1] = v
End
```

Pseudo Code - II

Student Name: ______________________________ J#: ________________

Q10) (15 pts) Consider an array A = [4, 8, 9, 1, 5, 3, 2, 6].

(a) Illustrate the sorting of the array using the Divide and Conquer approach of the Merge Sort algorithm.
(b) Determine the number of inversions in the array.

Show all the steps.