Ahmadu Bello University, Zaria
Department of Mathematics

2011/2012 First Semester Examination
COSC 401 Introduction to Algorithms & Complexity Analysis

Date: March 2013 **Time Allowed**: 120 Minutes

Instructions:

1. Attempt ANY FOUR questions.
2. Write all your answers in the spaces provided on this Question Paper.

Student's Registration Number:…… **Signature**:……

Date of Examination:……**Time**:……

Scores:

Question	Maximum Scores	Scores Obtained
1	20	
2	20	
3	20	
4	20	
5	20	
6	20	
Total	**80**	

1. (**20 Marks**). Consider the following algorithm.

```
ALGORITHM GE(A[0..n − 1, 0..n])
//Input: An n-by-n +1 matrix A[0..n − 1, 0..n] of real numbers
for i ← 0 to n − 2 do
   for j ← i +1 to n − 1 do
       for k ← i to n do
             A[j, k] ← A[j, k] − A[i, k] ∗ A[j, i] / A[i, i]
```

   a. (**3 Marks**). For what value(s) of `i` will this algorithm perform the maximum number of iterations?

   b. (**10 Marks**). Determine the time efficiency class of this algorithm. Show all details of your answer.

c. (**7 Marks**). Explain how you can improve the efficiency of this algorithm.

2. (**20 Marks**). Consider the following recursive algorithm.

```
Algorithm Q(n)
//Input: A positive integer n
if n = 1 return 1
else return Q(n – 1) + 2 * n – 1
```

a. (**6 Marks**). Set up a recurrence relation for the number of multiplications made by this algorithm and solve it.

b. (**6 Marks**). Set up a recurrence relation for the number of additions/subtractions made by this algorithm and solve it.

c. (**7 Marks**). Set up a recurrence relation for this function's values and solve it to determine what this algorithm computes.

3. (**20 Marks**). Consider the following algorithm

```
Algorithm Question4(A[0..n-1],B[0..m-1])
for i ← 0 to n - m do
    j ← 0
    while j<m and B[j]=A[i+j] do
       j ← j+1
    if j=m return i
return -1
```

a. (**10 Marks**). Given that array A is of size n and array B is of size m

i. give an expression in terms of n and/or m for the minimum number of successful comparisons.

ii. give an expression in terms of n and/or m for the maximum number of successful comparisons.

b. (**10 Marks**). Given that array A is of size n and array B is of size m

i. give an expression in terms of n and/or m for the minimum number of unsuccessful comparisons.

ii. give an expression in terms of n and/or m for the maximum number of unsuccessful comparisons.

4. The following table shows a small instance of the Knapsack problem having five items with their respective weights and values.

Item No.	Item Weight	Item Value
1	7	₦42
2	3	₦12
3	4	₦40
4	5	₦25
5	2	₦35

Assuming a knapsack of capacity of 15 units of weight, answer the following questions.

a. (**5 Marks**). List the infeasible solutions, if any, for this instance of the problem along with their weights.

b. (**15 Marks**). List the best five solutions for this problem instance along with their weights and values.

5. **(20 marks).** Answer all the following questions.
    a. **(5 marks).** What is the smallest number of digits the product of two n-digit numbers can have?

    b. **(5 marks).** What is the largest number of digits the product of two n-digit numbers can have?

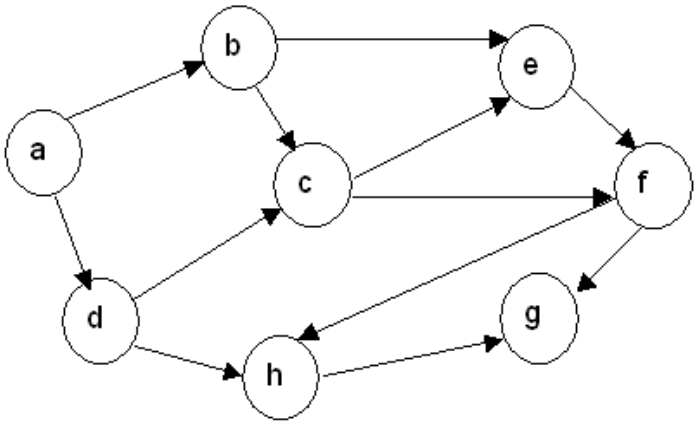
c. **(10 marks).** Consider the following algorithm

```
ALGORITHM Question6(A[0..n-1])
     //Input: An array A[0..n-1] of orderable elements
     //Output: Array A[0..n-1] of processed elements
     for i ← 1 to n-1 do
          j ← i-1
          while j>=0 and A[j] > A[j+1] do
               swap(A[j],A[j+1])
               j   ←  j-1
```

Trace the behavior of the algorithm on the following input array by completing the passes below. If you need additional passes draw them.

45, 68, 90, 29, 34, 17

6. (**20 Marks**). Use the following graph to answer the questions below.



Complete the following table by writing the order of arranging the vertices according to the traversal/algorithm indicated. You **must** arrange the vertices of the graph in sorted order (i.e., dictionary order) as much as possible.

a. (2.**5 Marks**). Depth-first traversal (from vertex **a**)

b. (2.**5 Marks**). Breadth-first traversal (from vertex **a**)

c. (**5 Marks**). Topological ordering of vertices (from the appropriate vertex) using the source removal algorithm.

d. (**10 Marks**). Show that `log(n!)` is in `O(n log n)`.