Name: Sec./B.N.:

Benha University
Faculty of Engineering (at Shoubra)
Electrical Engineering Department
M.Sc. (Computer Systems Engineering**)**

Midterm Exam
Subject: Advanced Algorithms - CES 608
Date: Sat 01/04/2017
Duration: 1 hour

Attempt the following questions.

№ of Questions: 4 in 1 page(s)
Total Points: 20 (10 **Marks**)

Question 1: **(05** pts**)**

Consider the algorithm for the sorting problem that sorts an array by counting, for each of its elements, the number of smaller elements and then uses this information to put the element in its appropriate position in the sorted array:

```
Algorithm ComparisonCountingSort(A[0..n − 1], S[0..n − 1])
//Sorts an array by comparison counting
//Input: Array A[0..n − 1] of orderable values
//Output: Array S[0..n − 1] of A's elements sorted in nondecreasing order
for i ← 0 to n − 1 do
    Count[i] ← 0
for i ← 0 to n − 2 do
    for j ← i + 1 to n − 1 do
        if A[i] < A[j]
            Count[j] ← Count[j] + 1
        else Count[i] ← Count[i] + 1
for i ← 0 to n − 1 do
    S[Count[i]] ← A[i]
```

a) Apply this algorithm to sorting the list 60, 35, 81, 98, 14, 47. **(**01 pt**)**
b) Is this algorithm stable? **(**02 pts**)**
c) Is it in place? **(**02 pts**)**

Question 2: **(05** pts**)**

Solve the following recurrence relations.

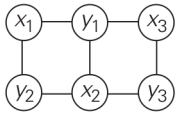
a) $x(n) = x(n-1) + 5$ for $n > 1, x(1) = 0$ **(**01 pt**)**
b) $x(n) = 3x(n-1)$ for $n > 1, x(1) = 4$ **(**01 pt**)**
c) $x(n) = x(n-1) + n$ for $n > 0, x(0) = 0$ **(**01 pt**)**
d) $x(n) = x(n/2) + n$ for $n > 1, x(1) = 1$ **(**01 pt**)**
e) $x(n) = x(n/3) + 1$ for $n > 1, x(1) = 1$ **(**01 pt**)**

Question 3: **(05** pts**)**

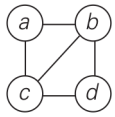
a) What is the efficiency of the brute-force algorithm for computing a^n as a function of n? As a function of the number of bits in the binary representation of n? **(**02 pts**)**
b) If you are to compute a^n mod m where a > 1 and n is a large positive integer, how would you circumvent the problem of a very large magnitude of a^n? **(**03 pts**)**

Question 4: **(05** pts**)**

A graph is said to be ***bipartite*** if all its vertices can be partitioned into two disjoint subsets X and Y so that every edge connects a vertex in X with a vertex in Y . (One can also say that a graph is bipartite if its vertices can be colored in two colors so that every edge has its vertices colored in different colors; such graphs are also called ***2-colorable***.) For example, graph (i) is bipartite while graph (ii) is not.



(i)



(ii)

a) Design a DFS-based algorithm for checking whether a graph is bipartite. **(**03 pts**)**
b) Design a BFS-based algorithm for checking whether a graph is bipartite. **(**02 pts**)**

Good Luck
Dr. Islam ElShaarawy