Benha University | ***Faculty of Computers & Informatics***

Final Exam
Class: *3rd* ***Year Students***
Subject: *Design and analysis of Algorithms*

Date: *10/1/2017*
Time: *3 hours*
Examiner: *Dr. El-Sayed Badr*

Answer the following questions:

Q1: (15 points) :

A) **Choose the correct answer of the following questions:**

نموذج الإجابة

1	**Sieve of Eratosthenes is an algorithm to** (A) The greatest common divisor (B) Prime numbers (C) The closest pair of points in plane (D) The shortest path in a graph (E) The inverse of a matrix
2	**Pseudocode may be defined as** a) Procedural solutions to problems b) A collection of connected geometric shapes containing a description of the algorithms steps c) A mixture of natural language and programming language like constructs d) A general approach to solving problems algorithmically e) Precise machine-readable description of an algorithm
3	**Algorithms that require ____________ number of operations are practical for solving only problems of very small size.** a) polynomial b) constant c) logarithmic d) linear e) exponential
4	**A function *t(n)* is said to be in *O(g(n))* if *t(n)*** a) is bounded above by *g(n)* for all large *n* b) is bounded above by some constant multiple of *g(n)* for all large *n* c) is bounded both above and below by some constant multiples of *g(n)* d) is bounded below by some constant multiple of *g(n)* for all large *n* e) is bounded above by some function of *g(n)*
5	**Brute force strategy of designing algorithms relies (depends) on using** a) the problem statements and definitions directly b) solution of a smaller instance of the same problem c) the combined solutions of smaller sub problems d) the solution to a simpler instance of the same problem e) the solution of an instance from a different problem

b)

Techniques of Algorithm	The problems which these techniques
1- Brute Force Technique	a) Knapsacke problem b) Bubble sort problem
2- Iterative technique	a) linear programming problems b) Mariage problem
3-Greedy Technique	a) Knapsacke problem b) minimum spanning tree
4- Divide and conquer technique	a) merge sort b) Matrix multiplication
5- Dynamic Programming Tech.	a) Knapsacke problem b)All-pairs shortest paths problem

Q2:

A)

Define the following terms:

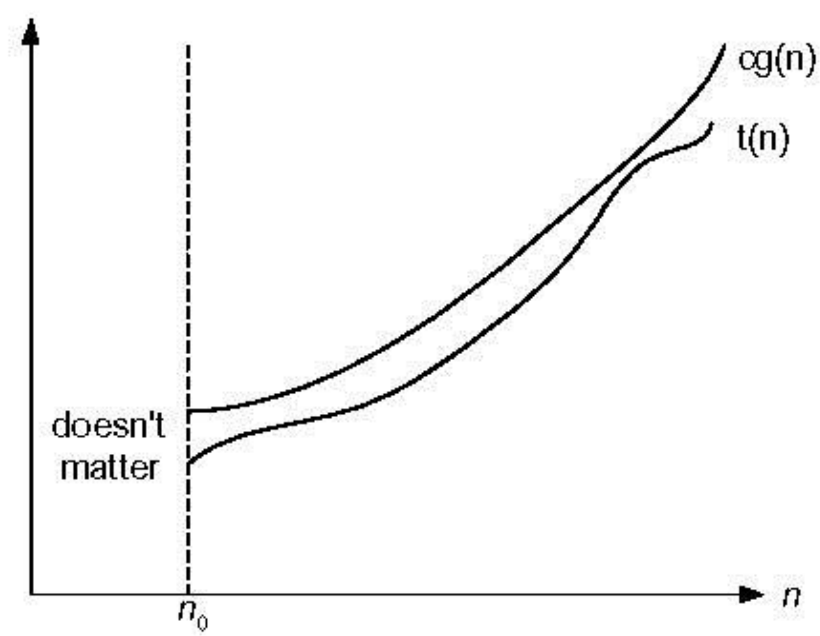
Big-oh Notation - Ω – Notation - Θ – Notation – Complexity of algorithm – sparse graph – dense graph.

Solution:

Definition:

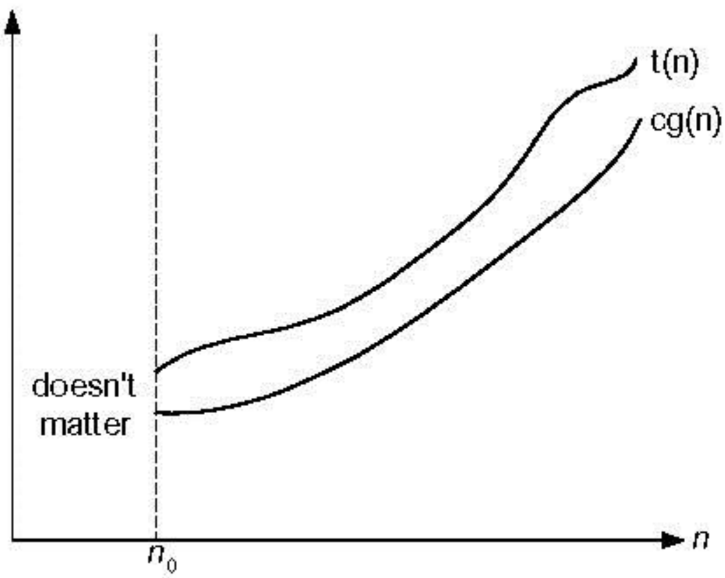
$f(n)$ is in O($g(n)$), denoted f$(n) \in O(g(n))$, if order of growth of $f(n) \leq$ order of growth of $g(n)$ (within constant multiple), i.e., there exist a positive constant c and non-negative integer n_0 such that

$$f(n) \leq c\, g(n) \text{ for every } n \geq n_0$$



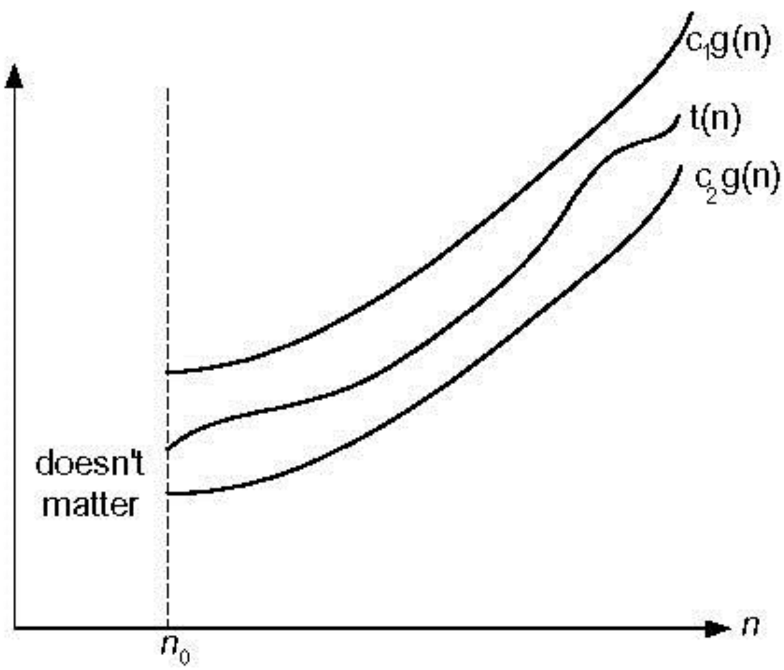
Definition

A function $t(n)$ is said to be in $\Omega(g(n))$, denoted $t(n) \in \Omega(g(n))$, if $t(n)$ is bounded below by some constant multiple of $g(n)$ for all large n, i.e., if there exist some positive constant c and some nonnegative integer n_0 such that t(n) $\geq$ cg(n) for all n $\geq$ n_0



Definition

A function *t(n)* is said to be in Θ*(g(n))*, denoted *t(n)* ∈ Θ*(g(n))*, if *t(n)* is bounded both above and below by some positive constant multiples of *g(n)* for all large *n*, i.e., <u>if there exist some positive constant c_1 and c_2 and some nonnegative integer n_0 such that</u> $c_2\, g(n) \le t(n) \le c_1\, g(n)$ for all $n \ge n_0$



Definition:

The number of instructions of an algorithm is called Complexity of the algorithm.

Definition:

A graph G is called sparse graph if $|E| < |V^2|$

A graph G is called sparse graph if $|E| \geq |V^2|$

Q2) <u>(15 points)</u>

b) Which is the best the adjecency matrix or the adjecency list for representing a given graph ?

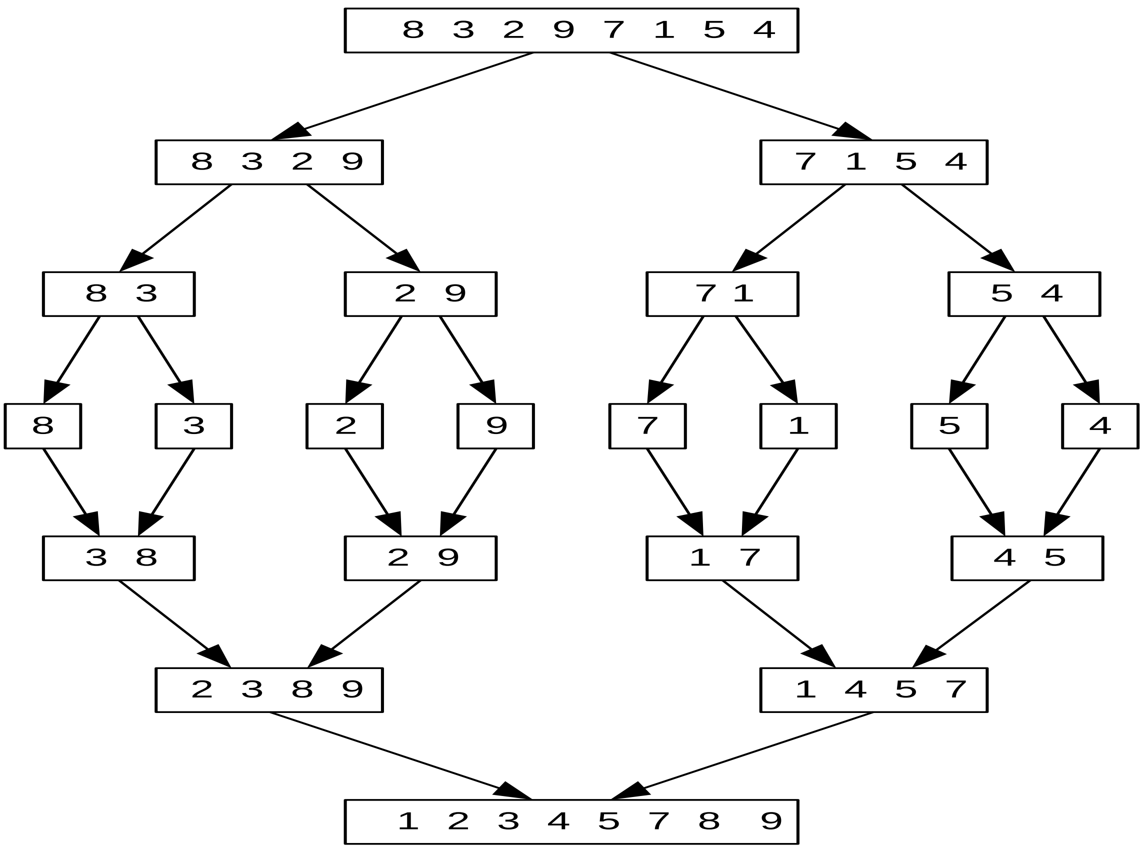
Solution:

a) The polynomial algorithm is the algorithm in which the number of instructions as polynomial function.
The exponentioal algorithm is the algorithm in which the number of instructions as exponential function.

b) Adjecency matrix is the best for dense graph and the adjacency list is the best for sparse graph.

Q3:

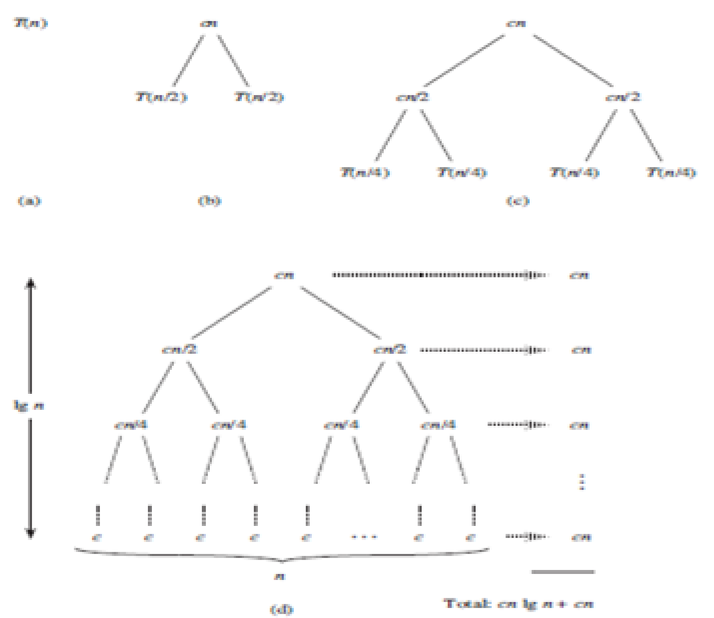
a)



Merge sort Pesudocod

```
MERGE(A, p, q, r)
n1 = q − p + 1
n2 = r − q
let L[1 .. n1 + 1] and R[1 .. n2 + 1] be new arrays
for i = 1 to n1
    L[i] = A[p + i − 1]
for j = 1 to n2
    R[j] = A[q + j]
L[n1 + 1] = ∞
R[n2 + 1] = ∞
i = 1
j = 1
for k = p to r
    if L[i] ≤ R[j]
        A[k] = L[i]
        i = i + 1
    else A[k] = R[j]
        j = j + 1
```

Analysis of Divided –Conquer Algorithms



Q4:

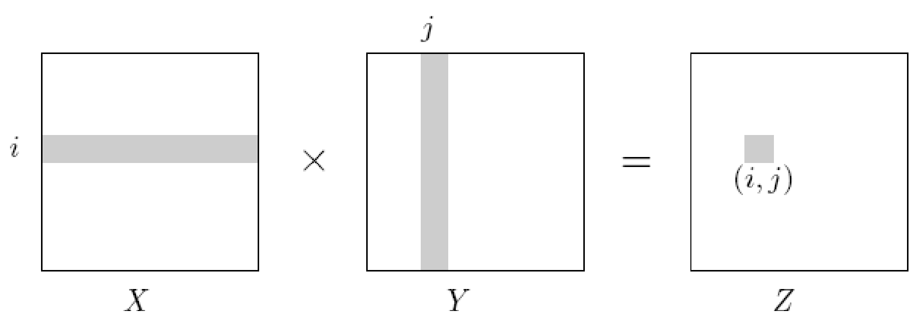
a)
first method

```
ALGORITHM  MatrixMultiplication(A[0..n − 1, 0..n − 1], B[0..n − 1, 0..n − 1])
    //Multiplies two n-by-n matrices by the definition-based algorithm
    //Input: Two n-by-n matrices A and B
    //Output: Matrix C = AB
    for i ← 0 to n − 1 do
        for j ← 0 to n − 1 do
            C[i, j] ← 0.0
            for k ← 0 to n − 1 do
                C[i, j] ← C[i, j] + A[i, k] ∗ B[k, j]
    return C
```

$$T(n) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\sum_{k=0}^{n-1} 1 = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} n = \sum_{i=0}^{n-1} n^2 = n^3 = \Theta(n^3)$$

Second method:

Suppose we want to multiply two matrices of size *N* x *N* : for example *X* x *Y* = *Z*. (*XY* *YX*)



$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad Y = \begin{bmatrix} E & F \\ G & H \end{bmatrix} \qquad XY = \begin{bmatrix} A & B \\ C & D \end{bmatrix}\begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}$$

We can use divide-conquer technique to compute XY by using recurrence relation 8 results with size n/2. It gives additional complexity $O(n^2)$ so $T(n) = 8\,T(n/2) + O(n^2)$

- Strassen showed that 2x2 matrix multiplication can be accomplished in 7 multiplication and 18 additions or subtractions. **.$(2^{\log_2 7} = 2^{2.807})$**
- This reduce can be done by Divide and Conquer Approach.

Divide and Conquer Matrix Multiply

$$A \times B = R$$

$$\begin{bmatrix} A_0 & A_1 \\ A_2 & A_3 \end{bmatrix} \times \begin{bmatrix} B_0 & B_1 \\ B_2 & B_3 \end{bmatrix} = \begin{bmatrix} A_0 \times B_0 + A_1 \times B_2 & A_0 \times B_1 + A_1 \times B_3 \\ A_2 \times B_0 + A_3 \times B_2 & A_2 \times B_1 + A_3 \times B_3 \end{bmatrix}$$

- Divide matrices into sub-matrices: A_0 , A_1, A_2 etc
- Use blocked matrix multiply equations
- Recursively multiply sub-matrices

Divide and Conquer Matrix Multiply

$$A \times B = R$$

$$\boxed{a_0} \times \boxed{b_0} = \boxed{a_0 \times b_0}$$

- Terminate recursion with a simple base case

Strassens's Matrix Multiplication

$$XY = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

$$P_1 = A(F - H) \qquad P_5 = (A + D)(E + H)$$
$$P_2 = (A + B)H \qquad P_6 = (B - D)(G + H)$$
$$P_3 = (C + D)E \qquad P_7 = (A - C)(E + F)$$
$$P_4 = D(G - E)$$

$$T(n) = 7T\left(\frac{n}{2}\right) + O(n^2) \rightarrow \quad O(n^{\log_2 7}) = O(n^{2.81}).$$

c) Prove that $T(n) = 6T(n/3) + n^2 \log n = \Theta(n^2 \log n)$ using Master's Theorem ?

Solution:

$$T(n) = 6T(n/3) + n^2 \log n$$
$$a = 6;\ b = 3;\ k = 2;\ p = 1$$
$$6 < 3^2 \rightarrow \quad a < b^k$$
$$3a \quad \rightarrow \quad T(n) = \Theta(n^2 \log n)$$

Q5:

Kruskal's algorithm

ALGORITHM *Kruskal(G)*

//Kruskal's algorithm for constructing a minimum spanning tree
//Input: A weighted connected graph $G = V, E$
//Output: ET, the set of edges composing a minimum spanning tree of G

sort E in nondecreasing order of the edge weights $w(e_1) \le \dots \le w(e_{|E|})$
$E_T \leftarrow \emptyset$; $ecounter \leftarrow 0$ //initialize the set of tree edges and its size
$k \leftarrow 0$ //initialize the number of processed edges
while $ecounter < |V| - 1$ **do**
    $k \leftarrow k + 1$
    if $E_T \cup \{e_{i_k}\}$ is acyclic
        $E_T \leftarrow E_T \cup \{e_{i_k}\}$; $ecounter \leftarrow ecounter + 1$
return E_T

b) We run the Kruskal's algorithm foe each component of the forest and choose the maximum weight for each step .

Good Luck