This question paper contains 4 printed pages]

BM—558—2013

FACULTY OF ENGINEERING

S.E. (CSE) EXAMINATION

MAY/JUNE, 2013

(New Course)

COMPUTER ALGORITHMS

(Saturday, 8-6-2013) **Time : 10.00 a.m. to 1.00 p.m.**

Time—Three Hours *Maximum Marks*—80

N.B.:— (*i*) *All* questions are compulsory.

(*ii*) Assume suitable data, if necessary.

(*iii*) Figures to the right indicate full marks.

Section A

1. Solve any *two* : 2×6=12

(*a*) Define computer algorithm. Explain Asymptotic notations θ, 0, Ω.

(*b*) State and explain Master theorem.

(*c*) Write an algorithm of merge sort and sort the given array :

{10, 1, 13, 21, 7, 9, 5, 15}.

P.T.O.

2. Solve any *two* : 2×7=14

(*a*) Determine upper bound for $T(n) = 4T(n/2) + n^2$. Verify using the substitution method.

(*b*) Write an algorithm for MAX_HEAPIFY procedure.

(*c*) Explain elements of Greedy method.

3. Solve any *two* : 2×7=14

(*a*) Find an optimal parenthesization of matrix chain product whose sequence of dimension is :

<30, 35, 15, 5, 10, 20>.

(*b*) Compare Divide and Conquer method and Dynamic Programming.

(*c*) Determine the LCS for :

x = {A, B, C, B, D, A, B}

y = {B, D, C, A, B, A}

Section B

4. Solve any *two* : 2×6=12

(*a*) Find the Huffman codes for the following set of sequences :

f : 5, e : 9, c : 12, b : 13, d : 16, g : 45

(*b*) Explain assembly line scheduling with example.

(*c*) Suppose that connected components on Undirected Graph G = (V, E), where;

$$V = \{a, b, c, d, e, f, g, h, i, j, k\}$$

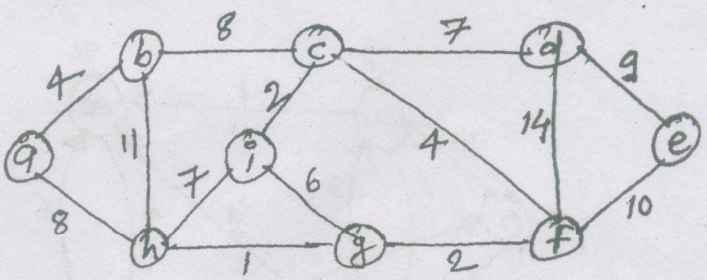
and edges are proceed in order :

(d, i), (f, x), (g, i), (b, g), (a, h), (i, j), (d, k), (b, j), (d, f), (g, j), (a, e) and (i, d).

Find the convected components of graph.

5. Solve any *two* : 2×7=14

(*a*) Construct MST using Prim's for given graph.



(*b*) Define Reducibility. Explain Cook's theorem.

(*c*) Write Bellman-Ford algorithm.

P.T.O.

6. Solve any *two* : 2×7=14

(*a*) Find OBST using dynamic programming set of keys $n = 4$:

qi	*pi*
0.5	—
0.15	0.10
0.05	0.10
0.05	0.05
0.3	0.1

(*b*) Define :

(*i*) Class P and NP.

(*ii*) NP-hard problems.

(*c*) Execute Dijkstra's algorithm for given graph.