Course Code : CST 201 / CST 213 **ITSJ/RW – 17 / 1017**

Third Semester B. E. (Computer Science and Engineering) Examination

DATA STRUCTURE AND PROGRAM DESIGN

Time : 3 Hours] [Max. Marks : 60

Instructions to Candidates :—

(1) Attempt All questions.
(2) All questions carry marks as indicated against them.
(3) Due credit will be given to neatness and adequate dimensions.
(4) Assume suitable data and illustrate answers with algorithms and sample execution trails wherever necessary.
(5) Mobile phones and / or electronic gadgets are prohibited in the examination hall.
(6) Use of non – programmable calculator is permitted.

1. Attempt any **Two** of the following :—

(a) Elaborate on the significance of time complexity of an algorithm. State [definitions] with examples the Bog – Oh [O], Theta [θ] and Omega [Ω] notations. 5 (CO 1)

(b) Convert the expression, A ^ (B * C) – (D * (E / B) to its equivalent RPN form. Evaluate the obtained RPN expression for A = 2 , B = 3 , C = 2 , D = 8 and E = 10. Show the contents (stack frame) at each stage of evaluation. 5 (CO 1)

(c) For an array based implementation of a circular queue, write the C code or algorithmic pseudocodes for insert and delete operations on a circular queue.
For an array of size 5, show the circular queue contents for following sequence of operations – Delete ( ), Insert ( ), Insert ( ), Insert ( ), at Front ( ), Delete ( ), Insert ( ), Insert ( ), Insert ( ), Delete ( ), Insert ( ), Insert ( ), isFull ( ). The key values are inserted in following sequence — 66 , 55 , 44 , 33 , 22 , 11 , 99 and 88. 5 (CO 1)

2. Attempt any **Two** of the following :—

(a) Develop algorithms or C functions to insert a node in a singly linked linear list –

(1) at the beginning of the list, and

(2) at the end of the list. The list pointer and data value of the node are passed as parameters. Write an algorithm or C function to create a list using any of these algorithms / functions. 5 (CO 1)

(b) Develop C functions to implement a linked queue. Write a program using these functions to simulate a QUEUE ADT. The program should print queue contents as required by user. 5 (CO 1)

(c) Design a C function to insert a node as the last node of a circular linked linear list. The function returns the address of the updated list. Show step–by–step trace while constructing a circular linked list using this function. 5 (CO 1)

3. Attempt any **Two** of the following :—

(a) Devise a recursive algorithm to count internal nodes with exactly 2 descendants in a binary tree. You must also print the nodes. 5 (CO 2)

(b) Create a C function to insert a node in a BST. The function must return a BST tree pointer. Use this function to design other function to create a BST. Also create a function for in – order walk of a BST. 5 (CO 2)

(c) How does a B tree differ from a B+ Tree ? Construct a 2 – 3 – 4 tree for the values stored in a list, B = [59 , 11 , 51 , 88 , 56 , 67 , 26 , 73 , 84 , 48 , 99 , 35 , 81 , 92 , 25]. Show B tree at each stage. 5 (CO 1)

4. Attempt any **Two** of the following :—

(a) Use mid square method to store the list, H = { 853 , 951 , 125 , 342 , 396 , 456 , 765 , 901 , 567 , 547 } in a hash table of size 11. Enlist the merits and limitations of mid square method. 5 (CO 3)

(b) Explain open hashing with appropriate example. Enlist the advantages and limitations of open hashing. 5 (CO 3)

(c) Given the list, X = { 567 , 985 , 563 , 294 , 923 , 678 , 123 , 654 , 238 } and H (x) = x mod 13, construct the hash table using —

(1) separate chaining, and

(2) open addressing with linear probing. 5 (CO 3)

5. Attempt any **Two** of the following :—

(a) Write an algorithm for implementing breadth first traversal on an undirected graph. For a graph in Fig. 5 (a), employ BFS showing intermediate BFS trees.

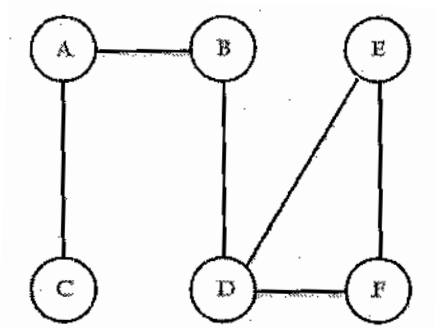

Fig. 5(a)

5 (CO 2 , 4)

(b) Write a C function or algorithmic pseudocode to exhibit Warshall's algorithm to find a shortest path matrix of a directed weighted graph. For a directed weighted graph in Fig. 5 (b), obtain the shortest path matrix showing intermediate steps.

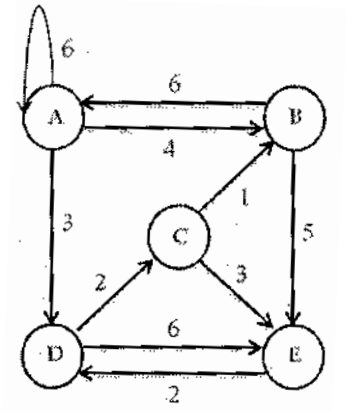

Fig. 5(b)

5 (CO 2 , 4)

(c) Write Prim's algorithm to construct a minimum spanning tree of a weighted graph. For a weighted graph in Fig. 5 (c), construct the MST and find its cost.

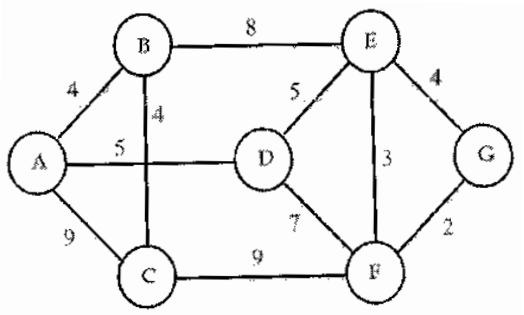

Fig. 5(c)

5 (CO 2 , 4)

6. Attempt any **Two** of the following :—

(a) Apply Heap sort on the list, H = { 55 , 88 , 22 , 99 , 77 , 33 , 11 } to order the list in ascending order. Draw heaps at the intermediate stages in building the maximal heap and during the stages of sorting process. 5 (CO 4)

(b) Elaborate on the characteristics bucket or radix sort. Develop an algorithm to implement bucket sort. Show step – by – step trace of bucket sort to order the list, B = { 1493 , 2812 , 1715 , 3710 , 4195 , 1437 , 2582 , 5340 , 1385 , 2090 }. 5 (CO 4)

(c) How does selection sort differs from bubble sort ? Write an algorithm or C function to order the list in increasing sequence using a bubble sort. Design an improved algorithm. BubbleImproved ( ), which guarantees that its best case time complexity is not quadratic. 5 (CO 4)