Course Code : CST 213 **GTHS/RS – 19 / 7148**

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

DATA STRUCTURE AND PROGRAM DESIGN

Time : 3 Hours] [Max. Marks : 60

Instructions to Candidates :—

(1) All questions carry marks as indicated against them.

(2) Illustrate answers with appropriate algorithm trace wherever necessary.

1. Attempt any **Two** :

   (a) Define a stack. List all allowable operations on a stack. Write C–functions to add a key to a stack and to remove a key from a stack. Write main() to test these functions appropriately. 5(CO1)

   (b) Explain the column–major ordering. Which programming languages use this ordering ? For A[6][7][3] with A[0][0][0] = 6000, compute the address of A[4][5][2] using row–and column–major ordering. 5(CO1)

   (c) What are advantages of using a circular queue ? Write an algorithm to remove an element from circular queue implemented using an array. Test your algorithm on an array, CQ[5] for all possible cases. 5(CO1)

2. Attempt any **Two** :

   (a) Consider a circular linked linear list. Write C–functions to –

      (1) Add a node to the list, and

      (2) Remove the last node of the list. Trace the algorithms or functions with appropriate example. 5(CO1)

   (b) Consider a singly linked linear list. Write C–functions to :

      (1) To create a list using element insertion at the end and,

      (2) To reverse the list without creating a new list. Show appropriate trace of your functions. 5(CO1)

(c) Write a menu driven C–program to implement a linked queue ADT. Ensure that all ADT operations are O(1). 5(CO1)

3. (a) Consider a binary search tree. Write algorithms or C–functions to :

(1) Create the tree,

(2) Destroy the tree, and

(3) Perform level–order traversal of the tree. Trace the routines appropriately. 5(CO2)

(b) Consider a binary tree. Write C–functions to :

(1) Copy the tree,

(2) Determine whether two trees are equal, and

(3) Count and print parent nodes with exactly two children.

5(CO2)

4. (a) Differentiate between open hashing and closed hashing [minimum 4 points]. For the keys – 54, 98, 34, 87, 29, 45, 29, 62, 37, 77 and 69, show the resulting hash table using open hashing. Let h(X) = X%11.

5(CO3)

(b) For the keys – 43, 71, 23, 61, 73, 99, 43, 44, 96, 79, 19 and 89. For a hash table with 13 buckets show the resulting :

(1) Open addressing hash table using sequential probing.

(2) Open addressing hash table using quadratic probing. 5(CO3)

5. Attempt any **Two** :

(a) For the directed graph in Fig. 5 [a], use Dijkstra's method to find shortest path tree starting at vertex B. Show the graph after each step.

5(CO2)

(b) For the weighted graph in Fig. 5[b], obtain the minimum cost spanning tree of the graph employing Prim's approach. Show the tree at each step.

5(CO2)

(c) Write an algorithm for implementing breadth first search on an undirected graph. For a graph in Fig. 5[b] employ BFS showing intermediate BFS trees. 5(CO2)

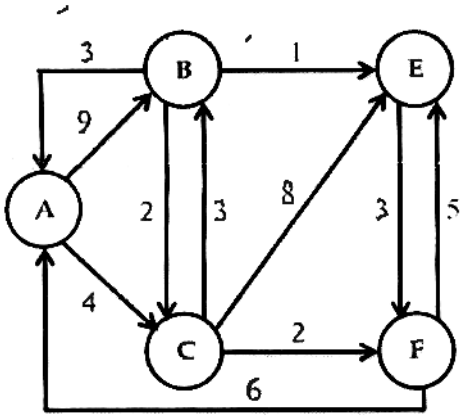
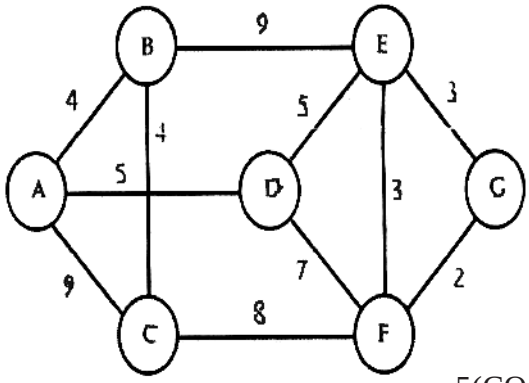


Fig. 5[a]



5(CO4)

Fig. 5[b]

6. Attempt any **Two** :

(a) Write an algorithm to order a list using Shell's method. Order the list L = {99, 88, 77, 66, 56, 44, 33, 22, 11}, using Shell's gap. 5(CO4)

(b) What do you understand by a stable sort and an in–place sort ? Execute merge sort on the list, L = {12, 97, 45, 36, 82, 31, 45, 76, 17}. Clearly show the call stack at each stage of distribution and integration. 5(CO4)

(c) How does a bucket sort differ from comparison–based sorts ? Implement bucket sort on the list, a Bucket [] = {444, 192, 934, 425, 678, 333, 582, 759, 892, 674, 555}.