Yashwant Singh

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
TEST-3 EXAMINATION- JUNE -2016
B.Tech. II Semester

COURSE CODE: 10B11CI211 MAX. MARKS: 35
COURSE NAME: Data Structures
COURSE CREDITS: 4 MAX. TIME: 2 HRS

Note: All questions are compulsory. Carrying of mobile phone during examinations will be treated as case of unfair means.

Q1. Consider the undirected graph given in Figure 1:
(a) Write BFS algorithm, and find a breadth-first spanning tree starting at A.
(b) Write Prim's algorithm, and find a minimum cost spanning tree by Kruskal's algorithm. **[2.5x2=5]**

Q2.(a) On which input data does the algorithm quick sort exhibit its worst-case behavior? Write a C program for insertion sort.
(b) Show all iterations, how quick sort sorts the following sequences of keys: 5, 5, 8, 3, 4, 3, 2. **[2.5x2=5]**

Q3.(a) Draw the 11-item hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16 and 5 using the hash function h (i) = (2i+5) mod 11. If collision occurs resolve it by chaining.
(b) How threaded binary tree is useful? Explain fully threaded binary tree by example. **[2.5x2=5]**

Q4.(a) Show the AVL tree that results after each of the integer keys 9, 27, 50, 15, 2, 21, and 36 are inserted, in that order, into an initially empty AVL tree. Clearly show the tree that results after each insertion, and make clear any rotations that must be performed.
(b) What is the maximum number of nodes in an AVL tree of a given height h? **[4+1=5]**

Q5. Consider the following array:

Index	0	1	2	3	4	5	6	7
values	unused	15	43	65	2	46	78	96

(a) Build the max heap, show the heap that results after each iterations.
(b) Write C code of heap sort algorithm and apply heap sort algorithm to above array show all iterations. **[2+3=5]**

Q6. Device a representation for a list where insertions and deletions can be made at either end. Such a structure is called a Deque (Double Ended Queue). Write C functions for inserting and deleting at either end. **[5]**

Q7. Two linked lists contain information of the same type in ascending order. Write a module to merge them to a single linked list that is sorted. **[5]**

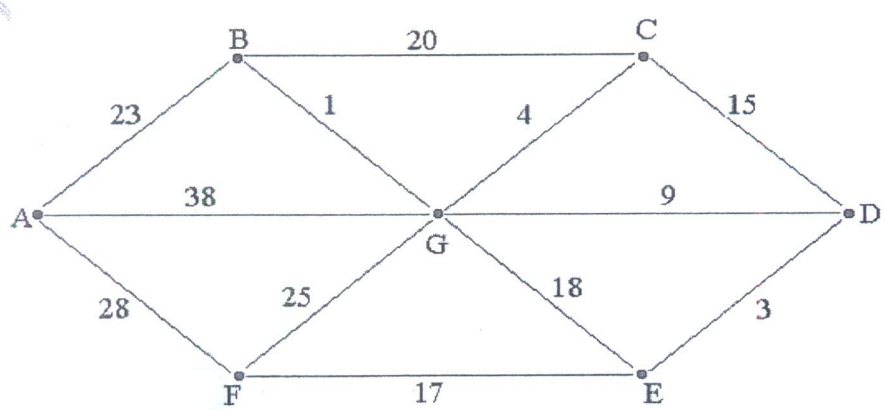

Figure 1