C 3141

B.E./B.Tech. DEGREE EXAMINATION, MAY/JUNE 2007.

Second Semester

(Regulation 2004)

Computer Science and Engineering

CS 1151 — DATA STRUCTURES

(Common to Information Technology)

(Common to B.E. (Part–Time) First Semester Regulation 2005)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Write the basic idea behind the Divide and conquer strategy.

2. What is O-notation?

3. What is an ADT?

4. List the characteristics of stacks.

5. What is a binary search tree?

6. What is hashing?

7. Develop pseudo code that will illustrate the process logic in Insertion sort.

8. What is the best case time complexity of the Quick sort algorithm?

9. What is an adjacency list? When is it used?

10. What is an activity node graph?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Outline in brief about Top–down design. (8)

(ii) Formulate an algorithm to convert a decimal integer to its corresponding octal representation. (8)

Or

(b) (i) Discuss in detail about the Implementation of algorithms. (8)

(ii) Given the character representation of an integer, convert it to its conventional decimal format. (8)

12. (a) (i) Write a program in C to return the position of an element X in a list L. (8)

(ii) State and explain the algorithm to perform Radix sort. (8)

Or

(b) (i) Write a program in C to create an empty stack and to push an element into it. (8)

(ii) Explain how queues can be implemented using Arrays. (8)

13. (a) (i) Construct an expression tree for the expression A + (B − C) *D + (E * F). (4)

(ii) Write a program in C to create an empty binary tree and to search for an element X in it. (12)

Or

(b) (i) Explain in detail the Linear probing technique. (8)

(ii) Write a function to delete the minimum element from a Binary heap. (8)

14. (a) State and explain the algorithm to perform Heap sort. Also analyze the time complexity of the algorithm. (16)

Or

(b) Write a C program to perform Merge sort. Also analyze the time complexity of the algorithm. (16)

15. (a) (i) Formulate an algorithm to find the shortest path using Dijkstra's algorithm. (8)

 (ii) Construct a Minimum Spanning Tree for the graph shown below. (8)
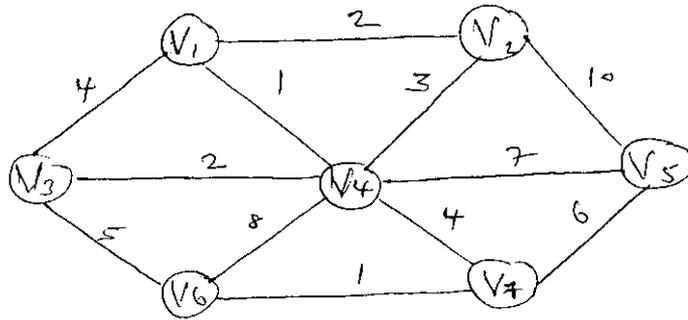
Or

(b) (i) Explain the Prim's algorithm with an example. (8)

 (ii) Write short notes on Biconnectivity. (8)