Reg. No. : □□□□□□□□□□□

P.S.G. LIBR

Question Paper Code : P 1198

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2009.

Third Semester

Computer Science and Engineering

CS 1151 — DATA STRUCTURES

(Common to Third Semester B.E.–Electronics and Communication Engineering and B.Tech.–Information Technology)

(Also Common to Second Semester B.E.–Computer Science and Engineering and B.Tech. Information Technology)

(Regulation 2004)

(Also Common to First Semester B.E. (Part-Time) Computer Science and Engineering – Regulation 2005)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Formally define *O* notation.

2. The worst case behaviour of merge sort and heap sort is $O(n \log n)$. However, heap sort is preferable than merge sort. Why?

3. Define NP-Complete.

4. What is the advantage of balancing height of any search tree?

5. When do you say a graph is bi-connected?

6. How does a search differ from a traversal?

7. What are the operations performed on a priority queue?

8. What is the advantage of top-down design?

9. What do you mean by abstract data type?

10. What is time-space trade-off?

PART B — (5 × 16 = 80 marks)

11. (a) Explain the Top-down design and implementation with example.

Or

(b) Assuming a sample algorithm analyse its efficiency in detail.

12. (a) (i) Describe List ADT. Using List ADT, explain the implementation of Stack ADT. (12)

(ii) What is the advantage of using list than array while implementing stack? (4)

Or

(b) (i) Design algorithms for various operations performed on circular linked list. (8)

(ii) Extend the algorithms defined in the previous question for the doubly linked circular list. (8)

13. (a) (i) Explain the four rotations performed on an AVL tree. (8)

(ii) Describe an algorithm for insertion of a key value into an AVL tree. (8)

Or

(b) What are the various collision resolution strategies in Hashing? Explain each one of them, and illustrate with examples.

14. (a) (i) Describe using an algorithm, how a pivot element be fixed in the appropriate position in the quick sort method? (8)

(ii) Derive the worst case behaviour of quick sort technique. (8)

Or

(b) (i) What is the need for external sorting? (4)

(ii) Explain any one of the method to perform external sorting. (12)

15. (a) Explain single source shortest path and all pair shortest, path problems with required algorithms and their complexities.

Or

(b) Explain with algorithm, how DFS be performed on an undirected graph. Then, show the algorithm for finding connected components of an undirected graph using DFS, and derive the time complexity of the algorithm.