T 8115

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2006.

Third Semester

Computer Science and Engineering

CS 1201 — DESIGN AND ANALYSIS OF ALGORITHMS

(Common to B.E. P.T. Second Semester Regulation 2005)

(Regulation 2004)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — ($10 \times 2 = 20$ marks)

1. What is an algorithm design technique?

2. Compare the order of growth $n!$ and 2^n.

3. What is the tool for analyzing the time efficiency of a non recursive algorithm?

4. What is algorithm animation?

5. Compare DFS and BFS.

6. Find the number of comparisons made by the sequential search in the worst and best case.

7. How efficient is Prim's algorithm?

8. What do you mean by Huffman Code?

9. What is state space tree?

10. What are the additional items required for branch and bound compare backtracking technique?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Discuss briefly the sequence of steps in designing and analyzing an algorithm. (10)

(ii) Explain some of the problem types used in the design of algorithm. (6)

Or

(b) (i) Explain the general frame work for analyzing the efficiency of algorithms. (8)

(ii) Explain the various asymptotic efficiencies of an algorithm. (8)

12. (a) (i) Design a recursive algorithm to compute the factorial function $F(n)=n!$ for an arbitrary non negative integer n and also derive the recurrence relation. (10)

(ii) Discuss the features of animation of an algorithm. (6)

Or

(b) (i) Design a non recursive algorithm for computing the product of two $n \times n$ matrices and also find the time efficiency of the algorithm. (10)

(ii) Write short notes on algorithm visualization and its applications. (6)

13. (a) (i) Write a pseudo code for divide and conquer algorithm for merging two sorted arrays into a single sorted one. Explain with an example. (12)

(ii) Set up and solve a recurrence relation for the number of key comparisons made by the above pseudo code. (4)

Or

(b) Design a recursive decrease–by–one algorithm for sorting the n real numbers in an array with an example and also determine the number of key comparisons and time efficiency of the algorithm. (16)

14. (a) (i) Construct a minimum spanning tree using Kruskal's algorithm with your own example. (10)

(ii) Explain single L–rotation and of the double RL–rotation with general form. (6)

Or

(b) Solve the all–pairs shortest–path problem for the diagraph with the weight matrix given below : (16)

	a	b	c	d
a	0	∞	3	∞
b	2	0	∞	∞
c	∞	7	0	1
d	6	∞	∞	0

15. (a) Apply backtracking technique to solve the following instance of the subset sum problem. $S = \{1,3,4,5\}$ and $d = 11$. (16)

Or

(b) Solve the following instance of the knapsack problem by the branch–and–bound algorithm. (16)

Item	Weight	Value
1	4	\$40
2	7	\$42
3	5	\$25
4	3	\$12

The Knapsack's capacity W = 10.