ROLL NO. ________________

Code: AC64/AT64 Subject: DESIGN & ANALYSIS OF ALGORITHMS

AMIETE – CS/IT (NEW SCHEME)

Time: 3 Hours **DECEMBER 2011** **Max. Marks: 100**

NOTE: There are 9 Questions in all.

- **Please write your Roll No. at the space provided on each page immediately after receiving the Question Paper.**
- **Question 1 is compulsory and carries 20 marks. Answer to Q.1 must be written in the space provided for it in the answer book supplied and nowhere else.**
- **The answer sheet for the Q.1 will be collected by the invigilator after 45 Minutes of the commencement of the examination.**
- **Out of the remaining EIGHT Questions, answer any FIVE Questions. Each question carries 16 marks.**
- **Any required data not explicitly given, may be suitably assumed and stated.**

Q.1 Choose the correct or the best alternative in the following: **(2×10)**

a. Given two sorted lists of size 'm' and 'n' respectively. The number of comparisons needed in the worst case by the merge sort algorithm will be:

(A) mn **(B)** max(m,n)
(C) min(m,n) **(D)** m+n–1

b. Which algorithm of matrix multiplication runs in $\theta\ (n^{lg7})$ time?

(A) Strassen's algorithm
(B) Matrix chain multiplication
(C) Naïve Matrix-multiplication algorithm
(D) None of the above

c. Fractional knapsack problem is solvable by:

(A) Greedy strategy **(B)** Dynamic programming
(C) Divide and conquer **(D)** None of the above

d. Prim's algorithm works on which approach?

(A) Greedy Strategy **(B)** Dynamic programming
(C) Divide and conquer **(D)** None of the above

e. Class NPC consists of those problems that are

(A) Solvable in polynomial time **(B)** Verifiable in polynomial time
(C) As hard as any problem in NP. **(D)** None of the above.

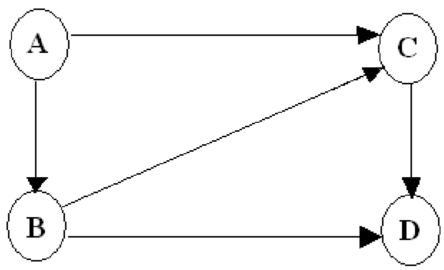
f. Which of the following does more work in solving the common sub-problems again and again?

(A) Dynamic programming does more work than divide and conquer strategy
(B) Divide and conquer does more work than dynamic programming
(C) The two approaches are not comparable
(D) Both does equal amount of work

g. The total running time of DFS is:

(A) θ (V + E) **(B)** θ (VE)
(C) θ (E lg V) **(D)** θ (V lg E)

h. Consider the graph below and find out valid topological sorting:



(A) A B C D
(B) B A C D
(C) B A D C
(D) A B D C

i. The matching time taken by KMP algorithm is:

(A) θ (n) **(B)** θ (m |∑|)
(C) O ((n – m + a)m) **(D)** θ (n^2)

j. Which one of the following in place sorting algorithms needs the minimum number of swaps?

(A) Quick Sort **(B)** Insertion Sort
(C) Selection Sort **(D)** Heap Sort

Answer any FIVE Questions out of EIGHT Questions.
Each question carries 16 marks.

Q.2 a. Briefly discuss all the asymptotic notations with examples. **(8)**

b. Solve the following recurrences
(i) T(n) = T(√n) + θ (lg lg n)
(ii) T(n)= 10T(n/3) + $17n^{1.2}$ **(8)**

Q.3 a. Write Merge Sort Algorithm. Prove that the running time complexity of Merge sort is O(n lg n). **(8)**

b. Explain Brute Force algorithm for string matching along with time complexity. **(8)**

Q.4 a. Give Breadth first traversal algorithm along with its complexity. **(8)**

b. Explain the following variable size decrease algorithms:
(i) Interpolation search
(ii) Searching and insertion in a binary search tree. **(8)**

Q.5 a. Explain various rotations used in AVL trees for balancing a tree. **(8)**

b. Illustrate the operation of Heapsort on the array A=< 5,13,2,25,7,17,20,8,4 > and show all the iterations involved in it. **(8)**

Q.6 a. Briefly discuss the Kruskal's algorithm for finding out the minimum spanning tree of a graph. Also analyse the time complexity of the Kruskal's algorithm. **(8)**

b. Give a Dynamic Programming solution for computing a binomial coefficient. Also discuss its time complexity. **(8)**

Q.7 a. Define B-Tree. Also prove that if $n \geq 1$, then for any n-keys B-Tree T of height h and minimum degree $t \geq 2$, then

$$h \leq \log_t \frac{n+1}{2}$$ **(8)**

b. Explain P, NP and NP complete problems. Give an example for each. **(8)**

Q.8 a. Consider a set of 4 objects placed on the shelf along with their values and weights given in a table below:

Item No(i)	Value of the Item(V_i)	Weight of the item(w_i)
1	$45	3 kg
2	$30	5 kg
3	$45	9 kg
4	$10	5 kg

The maximum weight of the Knapsack is 16. Solve the above problem by using Branch and Bound Technique to maximize the value contained in the Knapsack. **(8)**

b. Describe N-Queens problem in context with Backtracking and also write the algorithm. **(8)**

Q.9 a. Explain any four fundamentals used in algorithmic problem solving. **(8)**

b. Explain features of any **FOUR** of the following:
(i) Sorting (ii) Searching
(iii) String processing (iv) Graph problems
(v) Combinatorial problems (vi) Geometric problems. **(8)**