(D) Heap Sort is not a comparison based sorting algorithm

34. In a permutation a1.....an of n distinct integers, where $n > 100$, an inversion is a pair (ai, aj) such that i < j and ai > aj. What would be the worst case time complexity of the Insertion Sort algorithm, if the inputs are restricted to permutations of 1.....n with at most $100 * n$ inversions?
(A) $\Theta(n^2)$
(B) $\Theta(nlogn)$
(C) $\Theta(n^{1.5})$
(D) $\Theta(n)$

35. If Radix sort is used to sort an array of n integers which are in the range $(n^{log_2 d}, n^{d^2})$, where d is some function of input size, the time taken would be?
(A) $O(nd^2)$
(B) $O(n^2 d \log n + n^2 \log_2 d \log_2 n)$
(C) $O(nd^2 \log_2 n)$
(D) $O(n^2 d^2 + n^2 \log_2 n)$

36. The number of elements that can be sorted in $\Theta(n2^{n \log n})$ time using merge sort is
(A) $\Theta(n^{\log n})$
(B) $\Theta((\log n)^n)$
(C) $\Theta(\frac{n \log n * 2^{n \log n}}{(\log n)})$
(D) $\Theta(\frac{2^{n \log n}}{logn})$

7 Dynamic Programming

1. If an optimal solution can be obtained for a problem by obtaining optimal solutions for its sub-problems then the problem is said to have
(A) Optimal substructure property
(B) Regular property
(C) Overlapping sub-problems property
(D) Optimal sub-problems property

2. Instead of solving a problem directly if it can be broken into many smaller sub-problems which can be reused again and again, the problem is said to have:
(A) Optimal substructure property
(B) Regular property
(C) Overlapping sub-problems property

(D) Optimal sub-problems property

3. Instead of solving a problem directly if it can be broken in to many non-overlapping sub-problems and the solution to the original problem can be obtained by combining optimal solutions to these non-overlapping problems then such a strategy is called
(A) Dynamic programming
(B) Greedy
(C) Divide and conquer
(D) Recursion

4. Which of the following is true:
(A) If on some problem Dynamic programming is applicable, it takes less amount of time to solve it than other problem solving methodologies.
(B) Greedy method can solve a broader range of problems than than that of dynamic programming
(C)Since Greedy method is also based on optimal substructure property and overlapping subproblems property, solution provided by Greedy is always optimal solution.
(D) None of the above.

5. Which of the following problems can be solved using dynamic programming approach?
(A) Binary search
(B) Mergesort
(C) Longest common subsequence
(D) Quicksort

6. Which of the following problem solving technique can be employed to get the n^{th} term in the *Fibonacci* sequence:
(A) Dynamic programming
(B) Iterative approach
(C) Recursive approach
(D) All of the above

7. Four matrices M1, M2, M3 and M4 of dimensions pxq, qxr, rxs and sxt respectively can be multiplied is several ways with different number of total scalar multiplications.
For example, when multiplied as ((M1 X M2) X (M3 X M4)), the total number of multiplications is $pqr + rst + prt$. When multiplied as $(((M1XM2)XM3)XM4)$, the total number of scalar multiplications is $pqr + prs + pst$.
If $p = 20$, $q = 50$,$r = 10$, $s = 5$ and $t = 100$, then the number of scalar

multiplications needed is
(A) 21000
(B) 18500
(C) 17500
(D) 16800
(E) 19000
(F) 50000

8. Let A1, A2, A3, and A4 be four matrices of dimensions 10 x 15, 15 x 20, 20 x 5, and 5 x 10, respectively. The minimum number of scalar multiplications required to find the product A1A2A3A4 using the basic matrix multiplication method is:
(A) 5500
(B) 1800
(C) 2050
(D) 2750

9. If Brute force approach is used to find all the possible ways of multiplying the given set of n matrices the what is the time complexity of this approach?
(A) O(n^2)
(B) O(n^3)
(C) O(n^4)
(D) Exponential

10. What is the time and space complexity of the dynamic programming approach of the matrix chain multiplication problem?
(A) $O(n^2)$, $O(n^2)$
(B) $O(n^3)$, $O(n^2)$
(C) $O(n^4)$, $O(n^2)$
(D) Exponential, $O(n^2)$

11. The subset-sum problem is defined as follows.
Given a set of n positive integers, $S = \{a_1, a_2, a_3, \ldots, a_n\}$, and positive integer W, is there a subset of S whose elements sum to W?
A dynamic programming solution for solving this problem uses a 2-dimensional Boolean array, X, with n rows and $W + 1$ columns. $X[i, j], 1 \leq i \leq n, 0 \leq j \leq W$, is TRUE, if and only if there is a subset of $\{a_1, a_2, \ldots, a_i\}$ whose elements sum to j.
Which of the following is valid for $2 \leq i \leq n$, and $a_i \leq j \leq W$?
(A) $X[i, j] = X[i - 1, j] \vee X[i, j - a_i]$
(B) $X[i, j] = X[i - 1, j] \vee X[i - 1, j - a_i]$
(C) $X[i, j] = X[i - 1, j] \wedge X[i, j - a_i]$

(D) $X[i, j] = X[i-1, j] \wedge X[i-1, j-a_i]$

12. In the above solution which entry of the array X, if TRUE, implies that there is a subset whose elements sum to W?
(A) X[1,W]
(B) X[n,0]
(C) X[n,W]
(D) X[n-1,n]

13. A sub-sequence of a given sequence is just the given sequence with some elements (possibly none or all) left out. We are given two sequences X[m] and Y[n] of lengths m and n respectively, with indexes of X and Y starting from 0.
We wish to find the length of the longest common sub-sequence(LCS) of X[m] and Y[n] as l(m,n), where an incomplete recursive definition for the function l(i,j) to compute the length of The LCS of X[m] and Y[n] is given below:
l(i,j) = 0, if either i=0 or j=0
= expr1, if i,j > 0 and X[i-1] = Y[j-1]
= expr2, if i,j > 0 and X[i-1] != Y[j-1]

Where expr1 and expr2 are:
(A) expr1 ≡ l(i-1, j) + 1
(B) expr1≡ l(i, j-1)
(C) expr2≡max(l(i-1, j), l(i, j-1))
(D) expr2≡max(l(i-1,j-1),l(i,j))

14. Which of the following problems is equivalent to the 0-1 Knapsack problem?
(A) A bag can carry a maximum weight of W. There are N fragile items which have a weight of $\{w_1, w_2, w_3, \ldots, w_n\}$ and a value of $\{v_1, v_2, v_3, \ldots, v_n\}$. We need to choose the items to get maximum value.
(B) In an exam A can solve N questions. The questions take $\{t_1, t_2, t_3, \ldots, t_n\}$ hours and carry $\{m_1, m_2, m_3, \ldots, m_n\}$ marks respectively. Maximum allowed time for the exam is T hours. A can either solve a question or leave it as there is no partial marks given. A need to choose questions in such a way that score obtained is maximum.
(C) There are infinite coins of denominations $\{v_1, v_2, v_3, \ldots.., v_n\}$ and a sum S. The task is to find the minimum number of coins required to get the sum S.
(D) All of the above
(E) Only (B) and (C)

15. The time complexity of the brute force approach to solve the 0/1 - knapsack problem?
(A) O(nlogn)
(B) O(n!)
(C) O(2^n)
(D) O(n^{logn})

16. What is the time and space complexity of dynamic programming solution of the 0/1-Knapsack problem with n items and a maximum weight of W?
(A) O(Wlogn), O(Wn)
(B) O($nW \log n$), O(nW log n)
(C) O(nW), O(Wn)
(D) O(n^W), O(nW log n)

17. Consider two strings A = "qpqprrp" and B = "pqpqrpqr". Let x be the length of the longest common subsequence (not necessarily contiguous) between A and B and let y be the number of such longest common subsequences between A and B.
Then x + 10y=?
(A) 35
(B) 45
(C) 55
(D) 65

18. A knapsack can carry a maximum weight of 60. There are four unbreakable items with weights 10, 30, 50, 70 and profits 50, 120, 150, 420. What is the maximum value of the items you can carry using the knapsack?
(A) 170
(B) 200
(C) 420
(D) 360

19. Which of the following can be solved using Greedy approach?
(A) 0/1 knapsack problem
(B) Subset sum problem
(C) Fractional knapsack problem
(D) All pairs shortest path problem
(E) Activity selection problem
(F) Huffman coding

20. Which of the following can be solved using Dynamic Programming approach?
(A) 0/1 knapsack problem

(B) subset sum problem
(C) fractional knapsack problem
(D) all pairs shortest path problem
(E) Matrix chain Multiplication
(F) Longest common subsequence problem
(G) Tower of Hanoi problem
(H) Fibonacci sequence
(I) Huffman coding

21. Which of the following expression represents the total number of ways of parenthesizing an expression with n pairs of parentheses?
(A) $\binom{2n}{n}$
(B) $n!$
(C) $\frac{\binom{2(n-1)}{(n-1)}}{n}$
(D) $\frac{\binom{2n}{n}}{n+1}$

22. What is the time and space complexity of dynamic programming approach to solve the boolean parenthesizing problem?
(A) $O(n^2)$, $O(n^2)$
(B) $O(n^3)$, $O(n^2)$
(C) $O(n^4)$, $O(n^2)$
(D) Exponential, $O(n^2)$

8 Greedy Approach

1. Which of the following is true about Huffman Coding.
(A) Huffman coding may not return optimum encoding in all cases
(B) No code is prefix of any other code.
(C) Huffman Codes may be lossy in some cases
(D) All of the above

2. Suppose the letters a, b, c, d, e, f have probabilities $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$,$\frac{1}{16}$, $\frac{1}{32}$, $\frac{1}{32}$ respectively.
Which of the following is the Huffman code for the letter a, b, c, d, e, f?

(A) 0, 10, 110, 1110, 11110, 11111
(B) 11, 10, 011, 010, 001, 000
(C) 11, 10, 01, 001, 0001, 0000
(D) 110, 100, 010, 000, 001, 111

3. In the correct encoding of above question, what is the average length of huffman code for any character?