

1	<p>Write brief notes on</p> <ol style="list-style-type: none"> <li>Approximation Algorithms</li> <li>Genetic Algorithms</li> <li>NP Problems</li> <li>Lower Bound Theory</li> <li>Empirical analysis</li> <li>Characteristics of Good algorithm</li> <li>Memory issue in algorithm analysis</li> <li>Big Theta Notation</li> <li>Potential Method Analysis</li> <li>Asymptotic analysis</li> </ol>
2	<ol style="list-style-type: none"> <li>Write a program to implement the Merge Sort Algorithm using divide and conquer strategy.</li> <li>Write a program for optimal merge patterns using greedy programming. The program should take input as number of lists to be sorted with the number of elements in each list. The output should be the sequence in a particular order in which the lists are to be sorted.</li> </ol>
3	<ol style="list-style-type: none"> <li>Write an algorithm to solve matrix multiplication problem using dynamic programming. The problem contains n number of matrices to be multiplied. It matters that in which sequence these should be multiplied. Give the parenthesization for the efficient multiplication of matrices.</li> <li>Write an algorithm for finding minimum spanning tree using prim strategy.</li> </ol>
4	<p>a) Optimal (Polygon) Triangulation problem :</p> <p>We are given a convex polygon <math>P = \langle V_0, V_1, \dots, V_{n-1} \rangle</math> &amp; a weight function <math>w</math> defined on triangles formed by sides &amp; chords of <math>P</math>. The problem is to find a triangulation that minimizes the sum of the weights of the triangle in the triangulation. One weight function is <math>W(\Delta v_i v_j v_k) =  v_i v_j  +  v_j v_k  +  v_k v_i </math> Where <math> v_i v_j </math> is the Euclidean distance from <math>v_i</math> to <math>v_j</math>. Use Dynamic Programming.</p> <p style="text-align: center;">Or</p> <p>The first n cells of the array E contains integers sorted in increasing order. The remaining cells all contain some very large integer that we may think of as infinity (name it as maxint). The array may be arbitrarily large and you don't know n. Give an algorithm to find a position of a given integer <math>x</math> (<math>x &lt; \text{maxint}</math>) in array in <math>O(\log n)</math> time</p> <ol style="list-style-type: none"> <li>Write an algorithm for knight tour problem using backtracking on a 8*8 chessboard. Your algorithm should give the possible order of positions the knight will travel in order to complete the tour of the chessboard.</li> </ol>