

DESIGN AND ANALYSIS OF ALGORITHMS LAB

(Common to both CSE , IT, CSE(AI&ML),CSE(DS))

Course Code: 22CS1102

L	T	P	C
0	0	3	1.5

Course Outcomes: At the end of the Course the student shall be able to

CO1: Develop programs for sorting a given set of elements and analyse its time complexity. (L3)

CO2: Solve and analyse the problems using greedy methods. (L3)

CO3: Solve and analyse the problems using dynamic programming.(L3)

CO4: Apply backtracking method to solve various problems. (L3)

CO5: Apply branch and bound method to solve 0/1 knapsack problem. (L3)

LIST OF EXPERIMENTS:

(Any 12 experiments from the following list to be performed)

1. Implement Selection sort and find how many steps are required to sort 10 elements.
2. Implement and Analysis factorial of a number program using iterative and recursive methods.
3. Implement Insertion Sort and analyse the time complexity.
4. Given two strings, find the minimum number of edits required to convert one string to another.
5. Write a program to find the Greatest Common Divisor of two numbers using recursion and find how many steps are required to execute it.
6. Sort a given set of elements using the quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n (the number of elements in the list to be sorted) and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
7. Write a program to check whether a given graph is connected or not using the DFS method.
8. Apply Greedy method to compress the given data using Huffman encoding.
9. Implement fractional knapsack problem using Greedy Strategy.
10. Implement minimum spanning tree using Prim's algorithm and analyse its time complexity.
11. Apply dynamic programming methodology to implement 0/1 Knapsack problem.
12. Solve the longest common subsequence problem using dynamic programming.
13. Find the length of the longest subsequence in a given array of integers such that all elements of the subsequence are sorted in strictly ascending order.
14. Apply dynamic programming methodology to find all pairs shortest path of a directed graph using Floyd's algorithm.
15. Implement matrix chain multiplication and find the optimal sequence of parentheses.
16. Find a subset of a given set $S = \{s_1, s_2, \dots, s_n\}$ of n positive integers whose sum is equal to

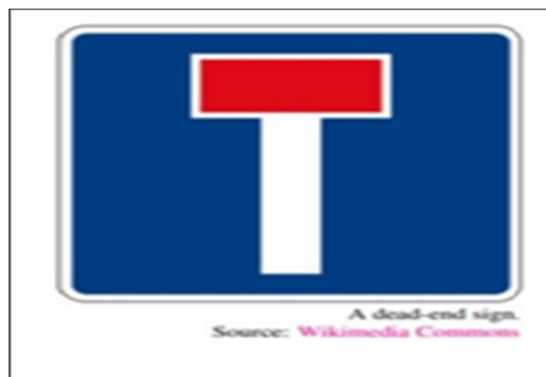
a given positive integer d . For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$ there are two solutions $\{1, 2, 6\}$ and $\{1, 8\}$. A suitable message is to be displayed if the given problem instance doesn't have a solution.

17. Implement N-Queens problem using backtracking.
18. Implement graph coloring problem using backtracking.
19. Find the solution of the 0/1 Knapsack Problem using LC Branch and Bound.
20. Find the solution to the Travelling Salesman Problem. Repeat the experiment for a graph having total number of nodes (n) = 4, 8, 12, 16, 20 and note the time required to find the solution. Plot the graph taking n on the x-axis and time on y-axis and analyze the graph to determine whether it is exponential or not.

21. Case Study 1:

The council of your home town has decided to improve road sign placement, especially for dead ends. They have given you a road map, and you must determine where to put up signs to mark the dead ends. They want you to use as few signs as possible. The road map is a collection of locations connected by two-way streets. The following rule describes how to obtain a complete placement of dead-end signs. Consider a street S connecting a location x with another location. The x -entrance of S gets a dead-end sign if, after entering S from x , it is not possible to come back to x without making a U-turn. A U-turn is

a 180-degree turn immediately reversing the direction. To save costs, you have decided not to install redundant dead-end signs, as specified by the following rule. Consider a street S with a dead-end sign at its x -entrance and another street T with a dead-end sign at its y -entrance. If, after entering S from x , it is possible to go to y and enter T without making a U-turn, the dead-end sign at the y -entrance of T is redundant. See Figure 1 for examples.



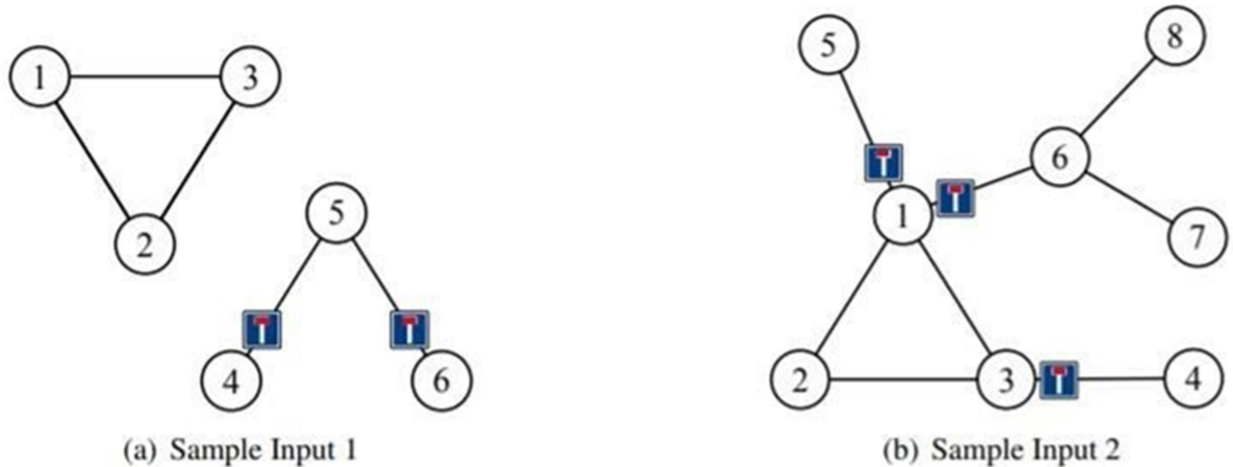


Figure 1: Illustration of sample inputs, indicating where non-redundant dead-end signs are placed.

Input:- The first line of input contains two integers n and m , where n ($1 \leq n \leq 5 \cdot 10^5$) is the number of locations and m ($0 \leq m \leq 5 \cdot 10^5$) is the number of streets. Each of the following m lines contains two integers v and w ($1 \leq v < w \leq n$) indicating that there is a two-way street connecting locations v and w . All location pairs in the input are distinct.

Output:- On the first line, output k , the number of dead-end signs installed. On each of the next k lines, output two integers v and w marking that a dead-end sign should be installed at the v -entrance of a street connecting locations v and w . The lines describing dead-end signs must be sorted in ascending order of v -locations, breaking ties in ascending order of w -locations.

22. Case Study 2:

The tropical island nation of Piconesia is famous for its beautiful beaches, lush vegetation, cocoa and coffee plantations, and wonderful weather all year round. This paradise is being considered as a future location for the World Finals of the ACM International Collegiate Programming Contest (or at the very least a vacation spot for the executive council). There is only one small problem: the island is really hard to reach. Currently, the fastest way to reach the island takes three days from the nearest airport, and uses a combination of fishing boat, oil tanker, kayak, and submarine. To make attending the ICPC World Finals slightly easier and to jump-start the island's tourism business, Piconesia is planning to build its first airport. Since longer landing strips can accommodate larger airplanes, Piconesia has decided to build the longest possible landing strip on their island. Unfortunately, they have been unable to determine where this landing strip should be located. Maybe you can help? For this problem we model the boundary of Piconesia as a polygon. Given this polygon, you need to compute the length of the longest landing strip (i.e., straight line segment) that can be built on the island. The landing strip must not intersect the sea, but it may touch or run along the boundary of the island. Figure 2 shows an example corresponding to the first sample input.

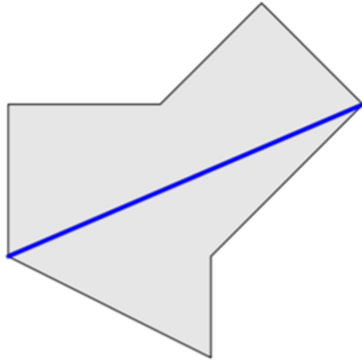


Figure 2: The island modeled as a polygon. The longest possible landing strip is shown as a thick line.

Input:- The input starts with a line containing an integer n ($3 \leq n \leq 200$) specifying the number of vertices of the polygon. This is followed by n lines, each containing two integers x and y ($|x|, |y| \leq 10^6$) that give the coordinates (x, y) of the vertices of the polygon in counter-clockwise order. The polygon is simple, i.e., its vertices are distinct and no two edges of the polygon intersect or touch, except those consecutive edges touch at their common vertex. In addition, no two consecutive edges are collinear.

Output:- Display the length of the longest straight line segment that fits inside the polygon, with an absolute or relative error of at most 10^{-6} .