| **WEEK-13** | **BACKTRACKING ALGORITHM** |
| --- | --- |
| 1. M coloring is a well known problem where, nodes of a given graph G = {V, E} is to be colored with maximum M colors such that any two nodes adjacent to each other are colored with different colors. We call the graph G as M colourable, if all nodes of the graph G can be colored using maximum M colors. Write a program (backtracking based) to find whether, the inputted graph G is M colourable (user inputted M) or not. 2. Indian intelligence department (IID) got confidential information about movement/journey of a terrorist from City A to City N (Fig. represents the road network of 14 cities, A to N). IID is sure that the terrorist just started his journey from City A to City N, but does not know that which path the terrorist is going to follow to reach at City N. So the IID planned to establish temporary check posts such that the terrorist can be caught before reaching to City N. Due to resource constraints, IID wants to establish minimum number of temporary check posts to catch the terrorist. Considering that the terrorist just started his journey from City A (i.e., he is yet to reach any one of the city next to City A in the paths from City A to City N), write a program to establish minimum number of temporary check posts to catch the terrorist before reaching to City N. In the example of below given figure, the minimum number of required check posts is 2, i.e. at City G and City H.      1. Travelling Salesman Problem (TSP) is a well known problem of a connected weighted graph (usually complete) G = {V, E}. In this problem, a salesman is to traverse all the cities (representing nodes of the graph) at least or at most once such that the total distance (weight of the edge between any two nodes represents the distance between the two nodes / cities) of the tour is least (minimum). In this tour, it is mandatory for the salesman to return back to the city / node from which journey was started. Below Fig. represents a sample connected weighted graph, G: Write a program (backtracking based) to find minimal distanced tour, if starting vertex V1 and a graph, G are given.     In Fig, if starting vertex as A, the tour is as follows: A->B->H ->E->F->G->D->C ->A and the distance traversed is 19.   1. Let us consider a network, G where, weights of edges represent the edge capacity or flow across the edges. For a given source, S and a tank, T in the network, G, let us consider, FG1 as the flow network and F as the maximum flow which you already computed between S and T.   (a) Suppose, G, FG1, and F are given to you. Considering that the capacity of a single edge in the given network, G is increased by 1, write an efficient program to compute the updated maximum flow F between source S and tank T without computing the maximum flow from scratch.  (b) Suppose, G, FG1, and F are given to you. Considering that the capacity of a single edge in the flow network, G is decreased by 1, write an efficient program to compute the updated maximum flow, f, without computing the maximum flow from scratch. | |