CityMate: Travel Management System

A Project Report Submitted

for the Course of

Minor Project - I
In

Third year – Fifth Semester of

Bachelor of Technology

In

Computer Science Engineering

With Specialization

In

Artificial Intelligence and Machine Learning
Under

Mrs. Sugandha Sharma

Associate Professor-

Department of Computer

Science

Prepared by

Specialization	SAP ID	Name
AI & ML	500075391	Abhijeet Jain
AI & ML	500076058	Aryan Verma
AI & ML	500075878	Chirag Sankhla

Department of Informatics
School Of Computer Science
UNIVERSITY OF PETROLEUM & ENERGY STUDIES,
DEHRADUN- 248007. Uttarakhand

INDEX

Project Title:

CityMate: Travel Management System

Abstract:

The traveling salesman problem is a permutation problem in which the goal is to find the shortest path between N different cities that the salesman takes is called the TOUR. In other words, the problem deals with finding a route covering all cities so that the total distance traveled is minimal. This paper gives a solution to find an optimum route for traveling salesman problem using Genetic algorithm technique, in which cities are selected randomly as initial population. The new generations are then created repeatedly until the proper path is reached upon reaching the stopping criteria

Introduction:

CityMate as the name suggests it has something to do with travelling in a city or travelling in any scope . For finding shortest distance between places , usually we go to google maps and get things done but What if ! we need various stops ,What if ! we need to plan our daily travels in the quickest way possible.This is where CityMate comes into play we worked on different algorithms to find the best way people didn't think they needed .

Literature Review:

The Travelling Salesman Problem (TSP) [1] is an optimization problem used to find the shortest path to travel through the given number of cities. Travelling salesman problem states that given a number of cities N and the distance between the cities, the traveler has to travel through all the given cities exactly once and return to the same city from where he started and also the cost of the path is minimized. This path is called as the tour and the path length is the cost of the path. In a complete weighted undirected graph G (V, E), cities are represented by the vertices and the distance between the cities are represented by the edges. The travelling salesman problem has to find the minimized Hamilton cycle that starts from some specified vertex, visits all other vertices exactly once return to the same specified vertex. The travelling salesman problem is used in many application areas [2] like planning, logistics, manufacturing of microchips, DNA sequencing, vehicle routing problems, robotics, airport flight scheduling, time and job scheduling of machines. The travelling salesman problem can be classified as Symmetric Travelling Salesman Problem (STSP), Asymmetric Travelling Salesman Problem (ATSP), and Multi Travelling Salesman Problem (MTSP). (i) STSP: In STSP the distance between two cities is same in both the directions that mean this will result in an undirected graph. (ii) ATSP: In ATSP the distance between two cities is not same in both directions. It is a directed graph and distance is different in both the directions. (iii) MTSP: In a given set of nodes, let there be 'm' salesmen located at a single depot node. The remaining nodes (cities) that are to be visited are intermediate nodes. Then, the MTSP finds the tours for all 'm' salesmen, who all start and end at the place, such that each intermediate node (city) is visited exactly once and the total cost of visiting all nodes is minimized. Solution of TSP may be of two types. The first will find the optimal solution which is almost nearer to the exact solution. This will guarantee the quality of the solution, but it is very slow. The second will find the nearest optimal solution within a reasonable time. It will not guarantee that the solution is nearer to the exact solution. So to improve the solution space and to increase the performance genetic algorithms are used to solve the travelling salesman problem which also gives the result within a reasonable amount of time..

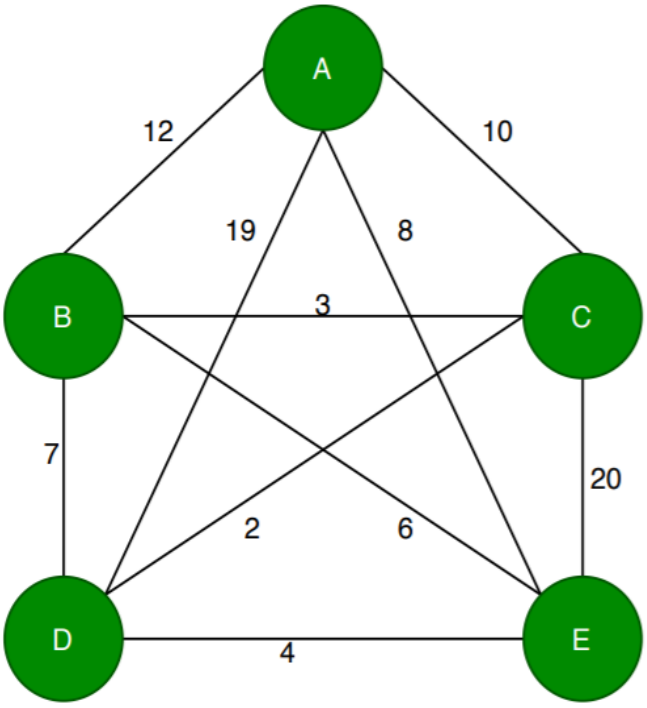
Objective:

- The objective of this project is to develop a console based C++ application named as City-Mate which finds the optimal route to travel between various cities (Locations).
- Understanding C++ language and Data-structures

Methodology:

In the TSP, given a set of cities and the distance between each pair of cities, a salesman needs to choose the shortest path to visit every city exactly once and return to the city from where he started.

Let's take an example to understand the TSP in more detail:



Here, the nodes represent cities, and the values in each edge denote the distance between one city to another. Let's assume a salesman starts the journey from the city . According to TSP, the salesman needs to travel all the cities exactly once and get back to the city by choosing the shortest path. Here the shortest path means the sum of the distance between each city travelled by the salesman, and it should be less than any other path.

With only cities, the problem already looks quite complex. As the graph in the example is a complete graph, from each city, the salesman can reach any other cities in the graph. From each city, the salesman needs to choose a route so that he doesn't have to revisit any city, and the total distance travelled should be minimum.

Algorithm:

Dijkstra Algo:

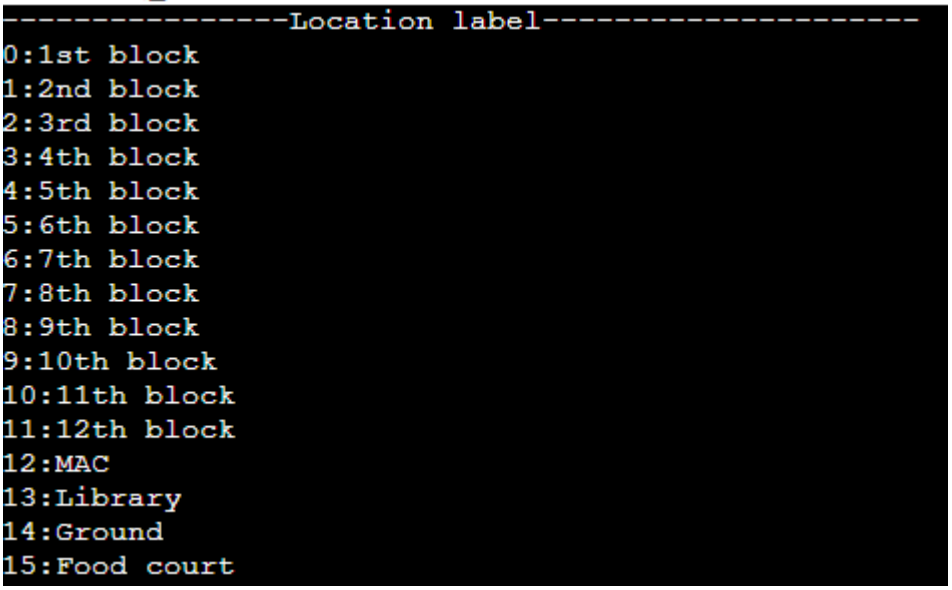
Step 1 : Passing the Adjacency matrix , source and destination to the function
Step 2: initializing distance array and visited array with int max and false
Step 3: setting distance of source equal to 0
Step 4: Starting a loop equal to number of the nodes
Step 5: Calculation of the temporary distances of all neighbour nodes of the active node by summing up its distance with the weights of the edges.
Step 6:calculated distance of a node is smaller as the current one, update the distance and set the current node as antecessor.
Step 7: add the node in path vector to display the path in order

Dynamic Programming :

Step 1: Send the source and Adjacency matrix to the function
Step 2: Create a array to mark the visited city
Step 3: all cities are unvisited, and the visit starts from the source city
Step 4: the TSP distance value is calculated based on a recursive function
Step5: if the distance is less than the minimum distance then added the distance as optimal cost and update the minimum distance
Step 6: Return the optimal cost

Result :

For testing the program is designed to work only for places that come under UPES campus .
The data of distance between different location is hard coded in the program.



```
Enter all the 4 location
Enter the Start point   1
2
0
Enter the Destination   3

-----------------------------------RESULT-------------------------
the Shortest path is of total--------12
2nd block->3rd block->4th block


The Path is:
2nd block--->1st block--->3rd block--->4th block--->1

Optimal cost is 40


...Program finished with exit code 0
Press ENTER to exit console.
```

Result Analysis:

In the dynamic algorithm for TSP, the number of possible subsets can be at most N x 2^N . Each subset can be solved in O(N) times. Therefore, the time complexity of this algorithm would be O(N^2 x 2^N) .

TSP is a popular NP-Hard problem, but depending on the size of the input cities, it is possible to find an optimal or a near-optimal solution using various algorithms.

k	algo	10	11	12	13	14	15	16	17	18	19	20
0	DP	1	2	4	7	17	44	1:55	4:60	12:60	33:10	1:24:45
0	A*	3	3	6	7	11	29	1:15	1:35.3	4:59	10:39	27:29
1	DP	2	3	8	20	59	2:55	8:37	24:36	1:08:52	3:18:19	9:09:31
1	A*	2	5	8	16	26.9	1:32	4:21	6:28	21:28	55:34	2:08:04
2	DP	2	5	16	52	2:50	9:32	31:06	1:42:58	5:25:04	*	*
2	A*	3	5	10	27	1:00	4:21	13:47	24:40	1:24:06	4:14:57	10:35:26
4	DP	3	9	38	3	13:07	1:00:28	4:29:48	*	*	*	*
4	A*	4	8	19	1:14	3:47	22:05	1:33:51	4:10:13	10:13:05	*	*
∞	DP	2	5	16	1:24	8:00	50:43	5:29:28	*	*	*	*
∞	A*	4	8	25	1:59	8:43	1:13:39	8:00:01	*	*	*	*

* the algorithm did not find a solution within 12 hours

The grey cells indicate the smallest average running time for a specific set of instances

Source : https://core.ac.uk/download/pdf/154419746.pdf

References:

GIT link : https://github.com/Abhijeet-try/CityMate

- See the TSP world tour problem which has already been solved to within 0.05% of the optimal solution. [1]
- ^ *Ross, I. M.; Proulx, R. J.; Karpenko, M. (6 May 2020). "An Optimal Control Theory for the Traveling Salesman Problem and Its Variants". arXiv:2005.03186 [math.OC].*
- Zomato solving travelling salesman problem https://zenodo.org/record/2656305/files/Online%20FDS.pdf
- Problem solving state representation https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/
- ^ A discussion of the early work of Hamilton and Kirkman can be found in *Graph Theory, 1736–1936* by Biggs, Lloyd, and Wilson (Clarendon Press, 1986).
- *Informed and Uninformed Search Algorithms https://www.geeksforgeeks.org/difference-between-informed-and-uninformed-search-in-ai/*
- ^ *Klarreich, Erica (8 October 2020). "Computer Scientists Break Traveling Salesperson Record". Quanta Magazine. Retrieved 13 October 2020.*
- ^ *Karlin, Anna R.; Klein, Nathan; Gharan, Shayan Oveis (30 August 2020). "A (Slightly) Improved Approximation Algorithm for Metric TSP". arXiv:2007.01409 [cs.DS].*
- Hash Map https://www.geeksforgeeks.org/hashing-data-structure/
- Niels Agatz, Paul Bouman, and Marie Schmidt. Optimization approaches for the traveling salesman problem with drone. Transportation Science, Forthcoming, 2017
- Richard Bellman. Dynamic programming treatment of the travelling salesman problem. Journal of the ACM (JACM), 9(1):61–63, 1962.
- https://core.ac.uk/download/pdf/154419746.pdf