VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

STADIUM TOUR MANAGEMENT SYSTEM

J-COMPONENT REPORT

Submitted by Team Members:-

Mitanshi Kshatriya (18BCE0540)

Kriti Chapagain (18BCE2483)

In the partial fulfillment for the award of degree of

Database Management Systems-CSE2004

under the guidance

of

Faculty: Govinda K

School of Computing Science and Engineering

ABSTRACT

The report is concerned with the travelling salesman problem, its theoretical approach, implementation and application.

TSP (Travelling Salesman Problem) is a typical issue of combinatorial optimization problem in the domain of mathematics, which aims at finding the shortest pathway among the given cities, and visits each city only once. This project introduces two ways to solve TSP based on genetic algorithm and Dynamic programming approach.

Genetic Algorithm is used to solve optimization problems and Travelling Salesman Problem (TSP) is an optimization problem. TSP is an NP hard problem in combinational optimization important in operations research and theoretical computer science. The amount of computational time to solve this problem grows exponentially as the number of cities. Dynamic programming is a useful mathematical technique for making a sequence of interrelated decisions. It provides a systematic procedure for determining the optimal combination of decisions. The main idea in this method's application is the division of a management process into several stages after which an optimal management model is chosen for each stage. The selected management model is the one which leads to optimal process functioning. The first algorithm finds one shortest path in a network with time dependent costs of links in O(knm) time, where n is the number of data, m is the number of links in the network. The second algorithm designed for the purpose of the project runs in O([n ^2]*2^n) time. Both algorithms are presented in the context of public transport networks and implemented using python and sqlite3.

The undertaking plans to facilitate the logistics required for conducting an arena tour. It intends to keep up all data like the staff alongside their expected set of responsibilities, urban communities to be visited. The diverse sort of types of gear required like speakers, mike, guitars, drum sets and so on together with their amounts alongside the various kinds of product that can be sold on area. The venture will utilize genetic calculation and dynamic programming approach for finding the shortest course to cover every one of the urban communities. This will make the stadium tour all the more economically plausible.

INTRODUCTION

The happening to web and empowering of downloading music, the music business is confronting a precarious decrease in CD sales which used to be fundamental earnings of music industry. The sales have endured to a great extent because of illicit downloads and membership based music administrations like Spotify, Ganna and so on. Presently the primary wellspring of salary of music industry is touring.

Because of poor logistics, blunder and poor route decisions numerous arena visits neglect to produce benefit. This task makes arena visiting a going great procedure and as practical as could reasonably be expected. The component of finding the most brief way will chop down the superfluous cost that are caused in transporting the stage equips just as the touring singer ,band individuals, back up artists, reinforcement artists, foundation vocalists and other required staffs.

This project has various modules that will help in betterment of logistics side of the tour. To generate shortest possible route consisting all required cities Genetic algorithm and dynamic programming approach is used. This problem concentrates on travelling salesman problem.

TSP is expressed as following: Let 1, 2, … , n be the marks of the n urban communities and C = [ci,j] be a n x n cost lattice where ci,j signifies the expense of heading out from city i to city j. TSP is the issue of finding the n-city shut visit having the base cost such that every city is visited precisely once. The absolute cost of a visit is:

A(n)= (i=1 to n-1)∑ $ci,i+1 + c1,n$

TSP is planned as finding a stage of cities, which has the base expense. This issue is known to be NP-hard. Numerous calculations have been proposed to tackle this issue. There are two fundamental approaches for illuminating TSP: precise and estimated. Precise approaches are generally founded on Dynamic Programming, Branch and Bound, Whole number Linear Programming… and all gave the ideal solutions for TSP.

Be that as it may, the calculations basing on these methodologies have exponential running time .Dynamic Programming takes O(n^2 .2^n) running time. Thus, they can just settle TSP with modest number of the vertices as calculations utilizing branch and bound strategy are just ready to give answers for 40 – 60 urban areas sets and ones utilizing direct programming fathom with most extreme for 200 cities sets.

Genetic algorithms, presented by J. Holland (1975), are motivated from the Darwin advancement hypothesis: in the populace development, the best people, which are more adjusted to their condition, can outlast for quite a while, on the on other hand, the people, which are not fits to their condition, vanish with the entry of ages. In this way, its chromosome and a proper wellness capacity to be characterized to assess people code every person. Initially, GA comprises to arbitrarily create introductory populace, at that point, genetic operators (selection, crossover, mutation), inside determined probabilities, are applied to deliver another generation that considered best than its previous versions.

LITERATURE REVIEW

A Stadium Tour Management framework is an attempt to defeat the logistics issue confronted while leading an arena visit. It will decrease the odds of scattering of any stage hardware as database can be utilized to cross check if each gear is available or not. Our venture would be a great guide to the tour manager as he/she will have all the data pretty much every one of the necessities of the visit. All the data can be seen by the administrator.

Different activities like include, erase, update and search are additionally present in the code. Aside from this all the arena subtleties alongside their ability and the tickets booked are put away in the framework.

The Stadium Tour Management framework has 4 modules:

1. Stadium

2. Staff

3. Equipments

4. Merchandise

All this module contains CURD features and some additional features.

PROPOSED WORK

The following project has used python 3.7 and sqlite3 package for database.

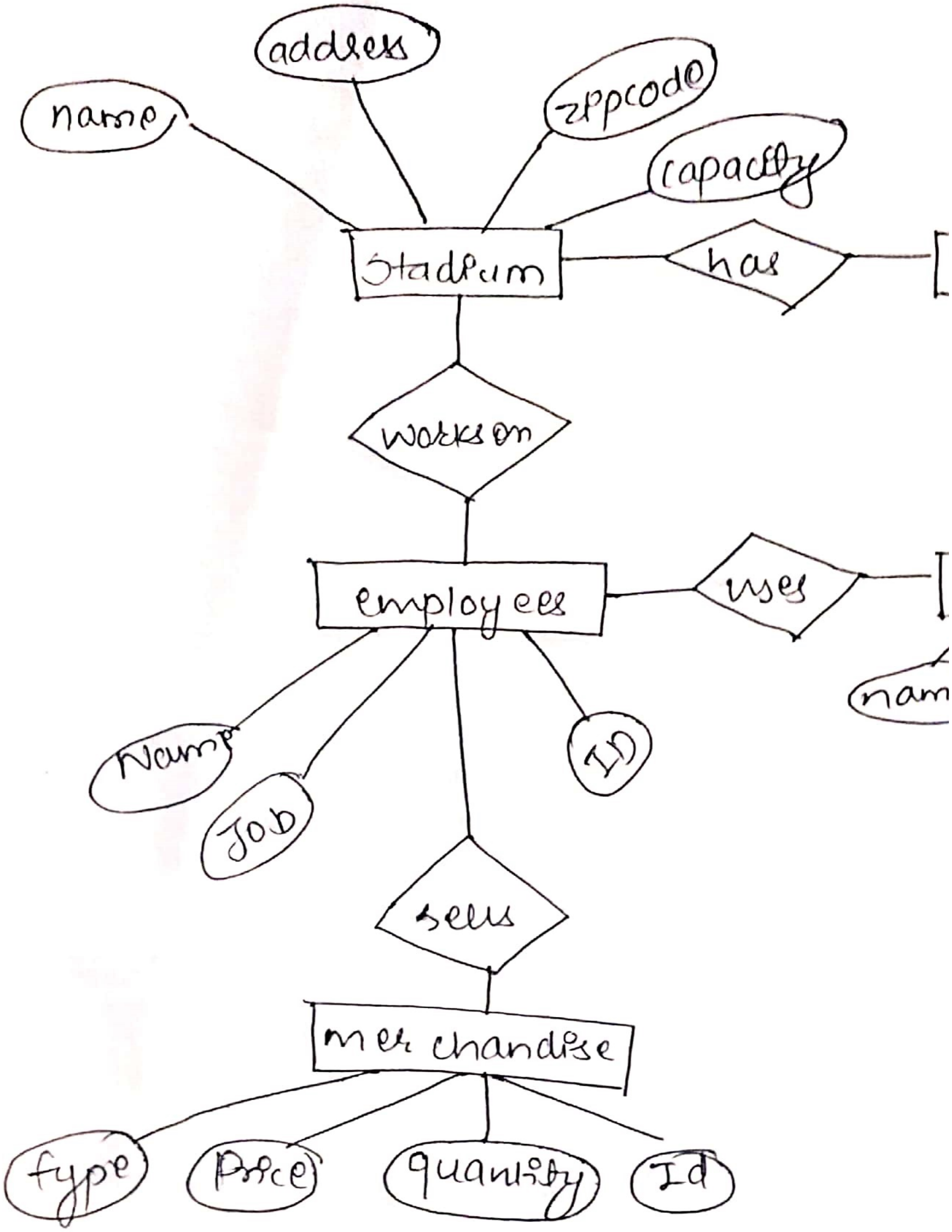
Generating the shortest route covering all cities is equivalent to travelling salesman problem. For this two approaches have been used-

1. Genetic algorithm

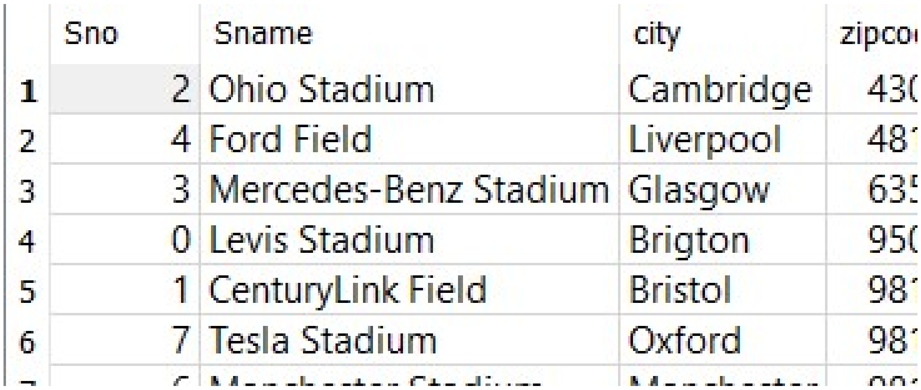
2. Dynamic programming

The distance matrix is generated with the help of a table containing distance of one city to other. Distance is zero for same city.

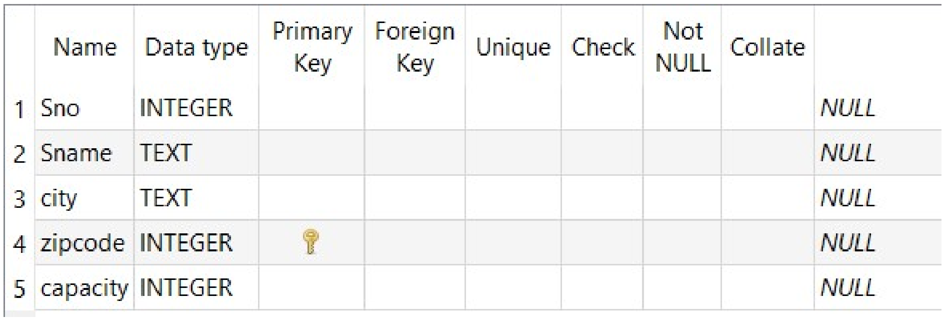
ER DIAGRAM:


address
name
zipcode
capacity
Stadium
has
works on
employees
uses
nam
Name
Job
ID
sells
merchandise
type
Price
quantity
Id

Stadium database

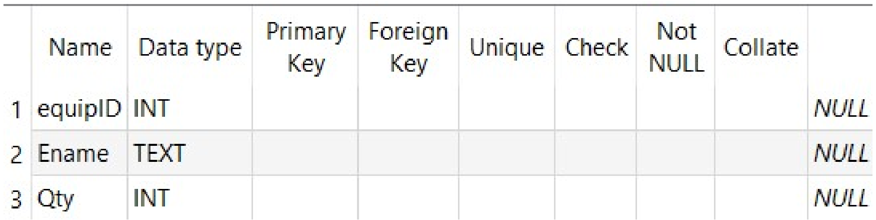


	Sno	Sname	city	zipco
1	2	Ohio Stadium	Cambridge	43
2	4	Ford Field	Liverpool	48
3	3	Mercedes-Benz Stadium	Glasgow	63
4	0	Levis Stadium	Brigton	95
5	1	CenturyLink Field	Bristol	98
6	7	Tesla Stadium	Oxford	98
7	6	Manchester Stadium	Manchester	98

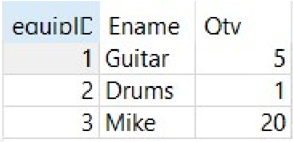


	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	Sno	INTEGER							*NULL*
2	Sname	TEXT							*NULL*
3	city	TEXT							*NULL*
4	zipcode	INTEGER	🔑						*NULL*
5	capacity	INTEGER							*NULL*

Equipment Database

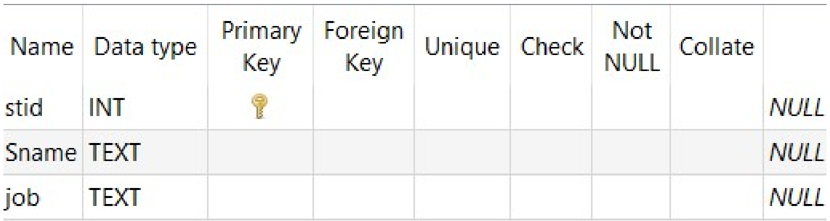


	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	equipID	INT							*NULL*
2	Ename	TEXT							*NULL*
3	Qty	INT							*NULL*

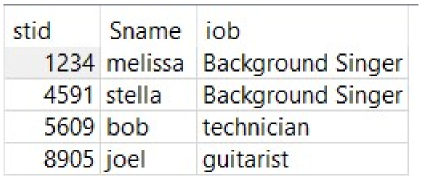


equipID	Ename	Qty
1	Guitar	5
2	Drums	1
3	Mike	20

Staff Database

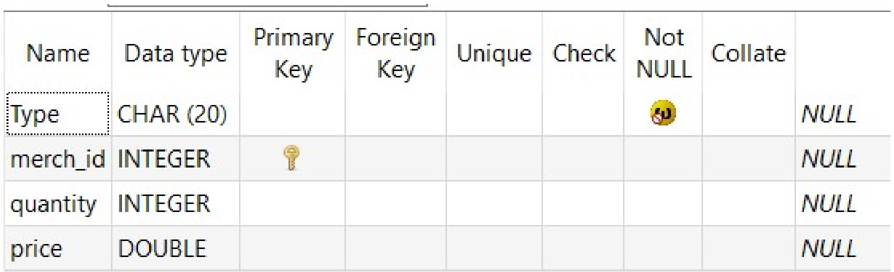


Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
stid	INT	🔑						*NULL*
Sname	TEXT							*NULL*
job	TEXT							*NULL*

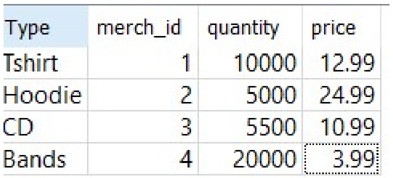


stid	Sname	iob
1234	melissa	Background Singer
4591	stella	Background Singer
5609	bob	technician
8905	joel	guitarist

Merchandise Database

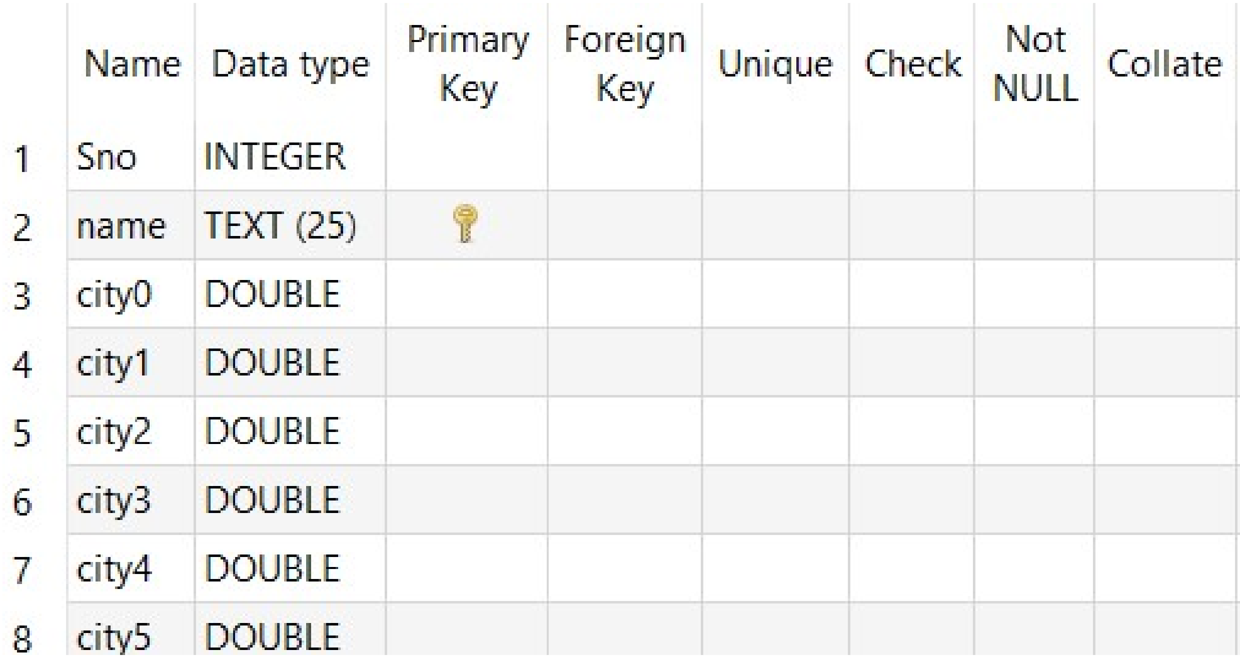


Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
Type	CHAR (20)					✓		*NULL*
merch_id	INTEGER	🔑						*NULL*
quantity	INTEGER							*NULL*
price	DOUBLE							*NULL*

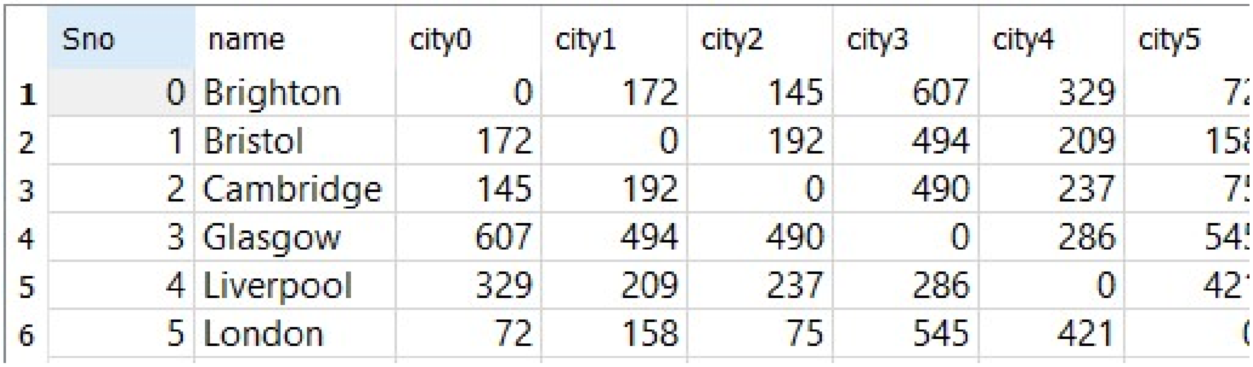


Type	merch_id	quantity	price
Tshirt	1	10000	12.99
Hoodie	2	5000	24.99
CD	3	5500	10.99
Bands	4	20000	3.99

Distance Matrix Table



	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate
1	Sno	INTEGER						
2	name	TEXT (25)	🔑					
3	city0	DOUBLE						
4	city1	DOUBLE						
5	city2	DOUBLE						
6	city3	DOUBLE						
7	city4	DOUBLE						
8	city5	DOUBLE						



	Sno	name	city0	city1	city2	city3	city4	city5
1	0	Brighton	0	172	145	607	329	7
2	1	Bristol	172	0	192	494	209	15
3	2	Cambridge	145	192	0	490	237	7
4	3	Glasgow	607	494	490	0	286	54
5	4	Liverpool	329	209	237	286	0	42
6	5	London	72	158	75	545	421	[illegible]

DBMS queries for creation of database-

tr.execute("CREATE TABLE stadium (Sno INTEGER,Sname TEXT,city TEXT,zipcode INTEGER PRIMARY KEY,capacity INTEGER);")

tr.execute("INSERT INTO stadium values(1,'Levis Stadium','Santa Clara',95050,107550);")

tr.execute("INSERT INTO stadium values(2,'CenturyLink Field','Seattle',98101,59677);")

tr.execute("INSERT INTO stadium values(3,'Ohio

```
Stadium','Columbus',43004,62897);")
tr.execute("INSERT INTO stadium values(4,'Mercedes-Benz
Stadium','Atlanta',63579,116746 );")
tr.execute("INSERT INTO stadium values(5,'Ford Field','Detroit',48127,49464);")

tr.execute("CREATE TABLE staff (stid INT PRIMARY KEY, Sname TEXT, job TEXT);")
tr.execute("INSERT INTO staff values(1234,'melissa','Background Singer');")
tr.execute("INSERT INTO staff values(4591,'stella','Background Singer');")
tr.execute("INSERT INTO staff values(5609,'bob','technician');")
tr.execute("INSERT INTO staff values(8905,'joel','guitarist');")

tr.execute("CREATE TABLE equipments (equipID INT, Ename TEXT, Qty INT);")
tr.execute("INSERT INTO equipments values(1,'Guitar',5);")
tr.execute("INSERT INTO equipments values(2,'Drums',1);")
tr.execute("INSERT INTO equipments values(3,'Mike',20);")

tr.execute("CREATE TABLE Merchandise (Type        CHAR (20) NOT NULL,
merch_id INTEGER      PRIMARY KEY,quantity INTEGER,price        DOUBLE);")
tr.execute("INSERT INTO Merchandise values('Tshirt',1,10000,12.99);")
tr.execute("INSERT INTO Merchandise values('Hoodie',2,5000,24.99);")
tr.execute("INSERT INTO Merchandise values('CD,3,5500,10.99);")
tr.execute("INSERT INTO Merchandise values('Bands',4,20000,3.99);")

tr.execute("Create table Dist2(name text,city0 double,city1 double,city2
double,city3 double,city4 double, city5 double, city6 double,city7 double,);")
```

Pseudo code for genetic algorithm

1. Creating distance matrix – It is a2D array consisting of a list of cities and distance between each pair of cities

2. Creating the population – Population is a collection of possible routes.

3. Determining the fitness of the population-

4 . Select the mating pool

5. Breed/ordered crossover

6. Mutate

7. Repeat the above process until the required number of iterations is reached. We are using 2000, 4000, 5000 iterations.

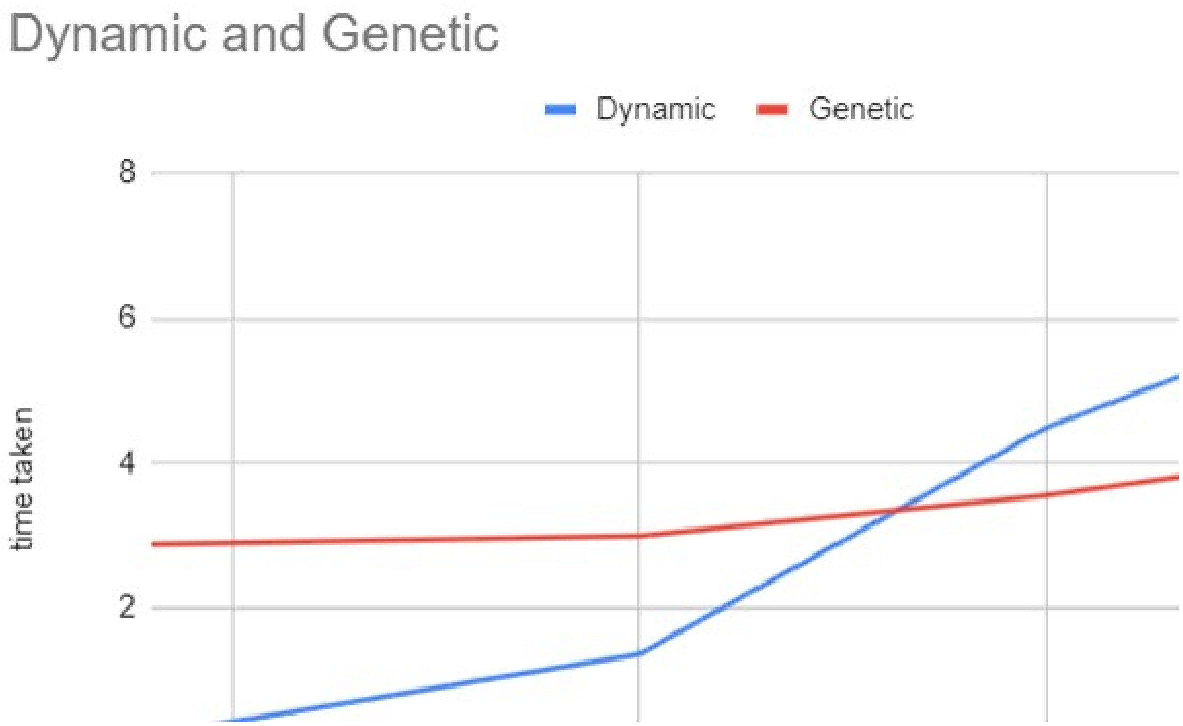
Pseudo code for dynamic programming

```
1. $Cost (\{1\}, 1) = 0$
2. for $s = 2$ to $n$ do
    2.1 for all subsets $S Є [1, 2, 3, … , n]$ of size $s$ and containing 1
        2.1.1 $C (S, 1) = \infty$
    2.2 for all j Є S and j ≠ 1
        2.2.1 Cost(S, j) = min [Cost (S – {j}, i) + d(i, j) for i Є S and i ≠ j]
3. Return minj Cost ({1, 2, 3, …, n}, j) + d(j, i)
```

RESULTS

Comparison of algorithm used-

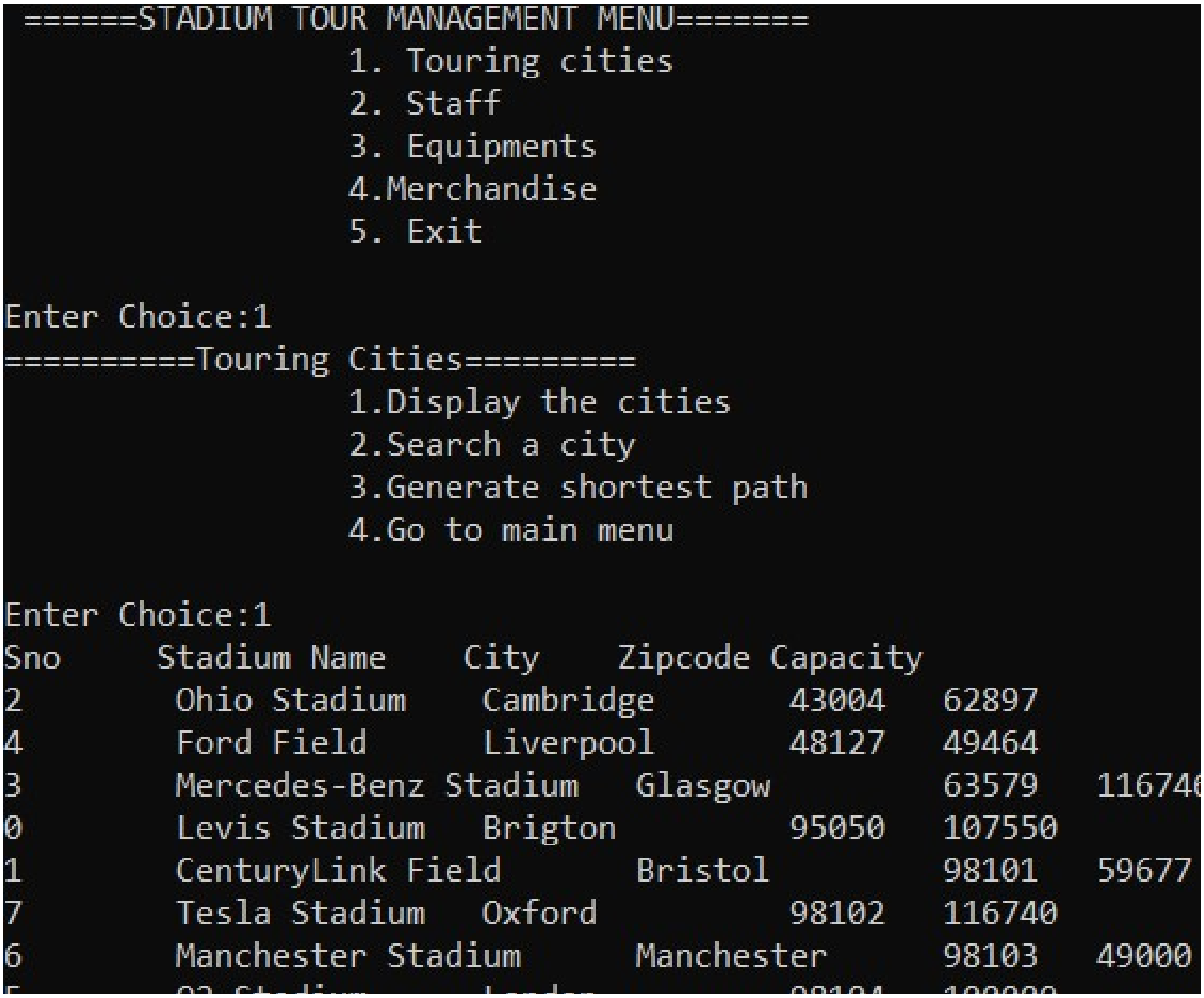
ALGORTIHM	Time Taken	Result(cost)	TIME COMPLEXITY	ADVANTAGE	DISADVANTAGE
GENETIC ALGORITHM	77.041	1355.0	O(Kmn)	Best possible solution is generated	No optimal solution is reached
Dynamic programming	0.2513	1355.0	$O(n^2 * 2^n)$	Generates a optimal solution Easy to code	Expensive for memory as well as time Quality of solution depends on the code



The accompanying chart shows that with less amount of data dynamic programming give a lot quicker outcome than hereditary calculation for tackling travelling salesman problem. However, as the size of information begins to expand the time taken by powerful programming builds exponentially when contrasted with that of genetic calculation.

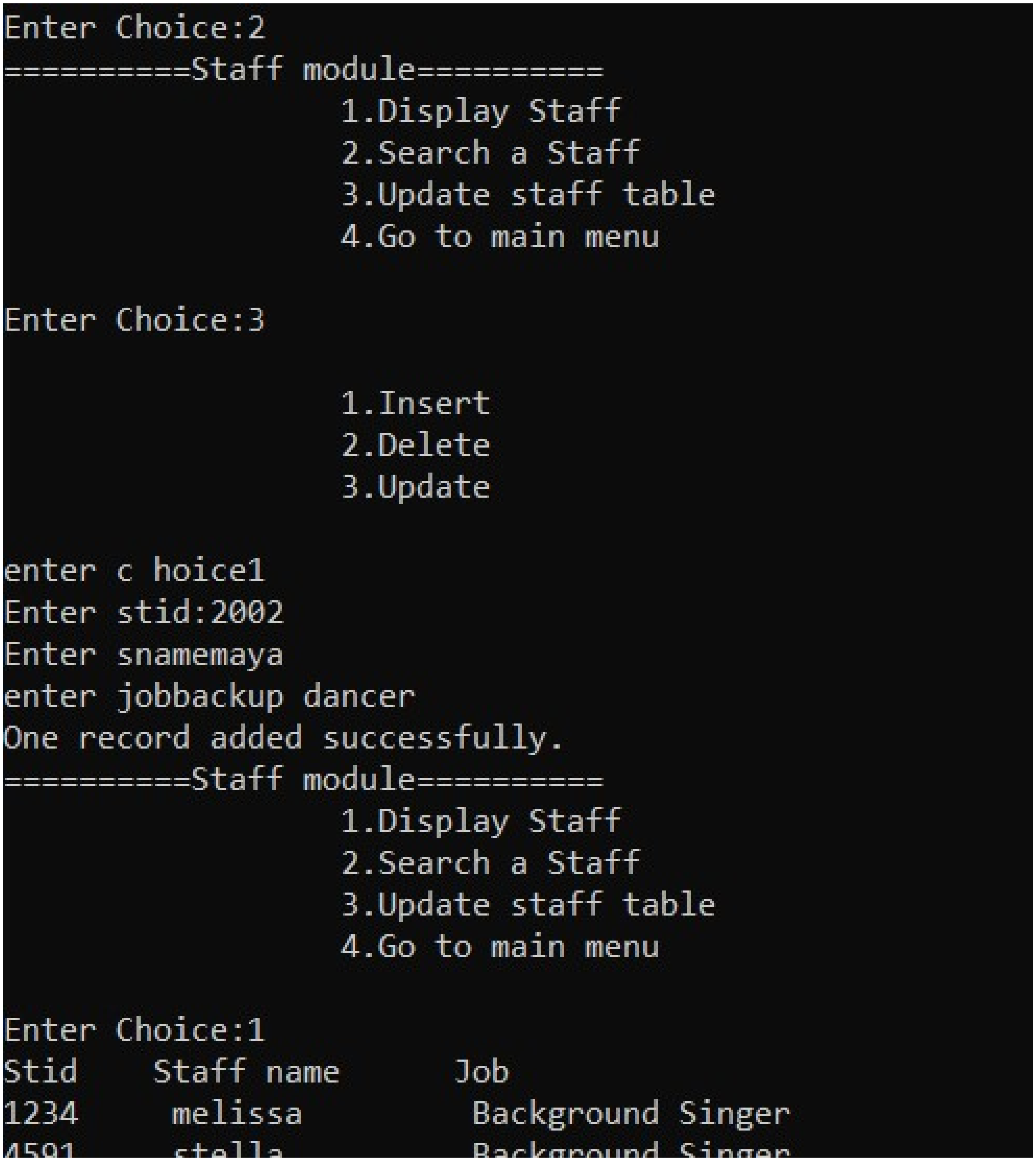
Genetic algorithm seem to discover great answers for the Traveling Salesman Problem, anyway it depends particularly on how the it is encoded and which crossover and mutation strategies are utilized. Likewise it doesn't have one optimal solution. However, it provides with best possible solution. Dynamic programming approach is relatively simple to code yet it requires some investment when the amount of data increments .Additionally, a beginning city must be indicated in its code.   It is expensive for memory as well as time. The quality of solution depends on the code.

Displaying Cities-



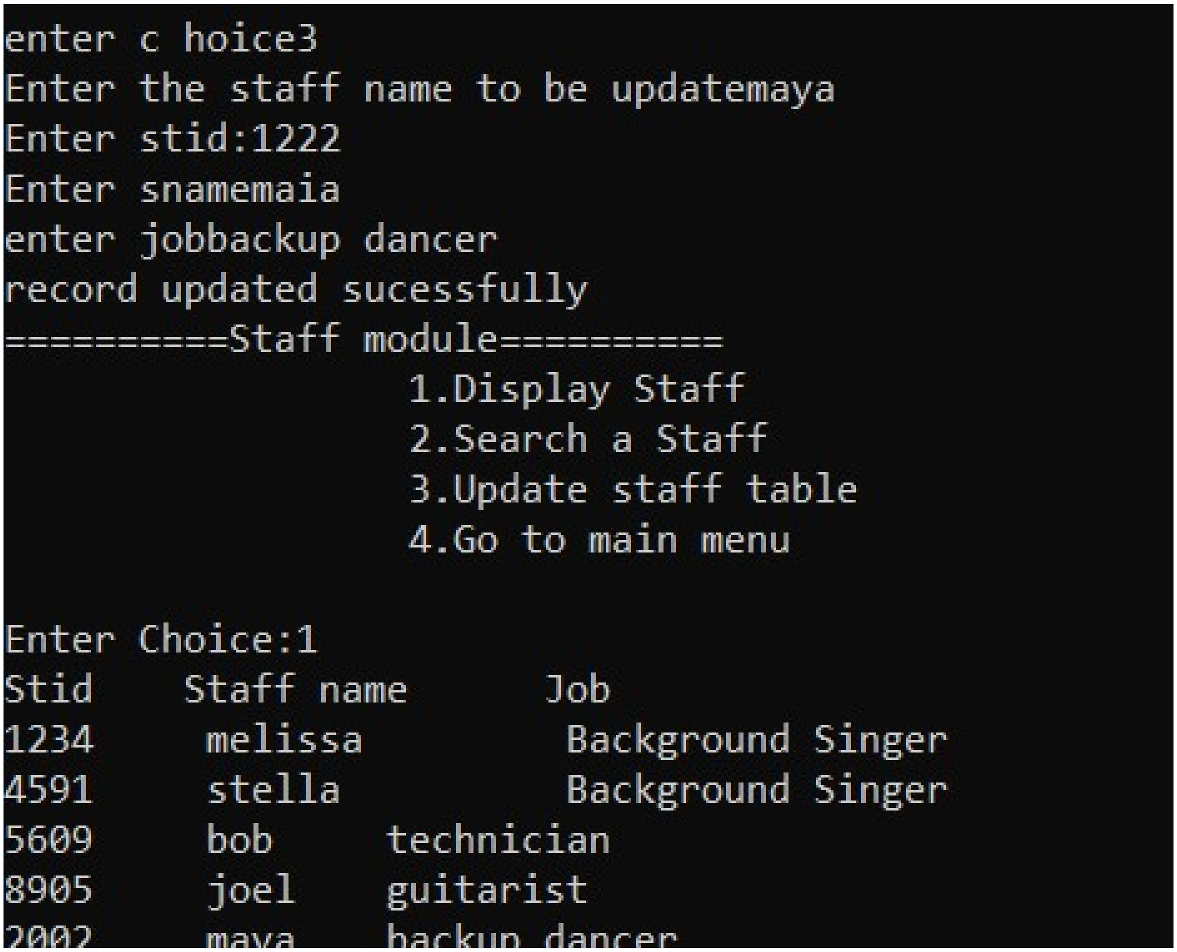
```
 ======STADIUM TOUR MANAGEMENT MENU=======
                  1. Touring cities
                  2. Staff
                  3. Equipments
                  4.Merchandise
                  5. Exit

Enter Choice:1
==========Touring Cities=========
                  1.Display the cities
                  2.Search a city
                  3.Generate shortest path
                  4.Go to main menu

Enter Choice:1
Sno     Stadium Name    City    Zipcode Capacity
2        Ohio Stadium    Cambridge       43004   62897
4        Ford Field      Liverpool       48127   49464
3        Mercedes-Benz Stadium   Glasgow         63579   11674
0        Levis Stadium   Brigton         95050   107550
1        CenturyLink Field       Bristol         98101   59677
7        Tesla Stadium   Oxford          98102   116740
6        Manchester Stadium      Manchester      98103   49000
```

Inserting in staff module-



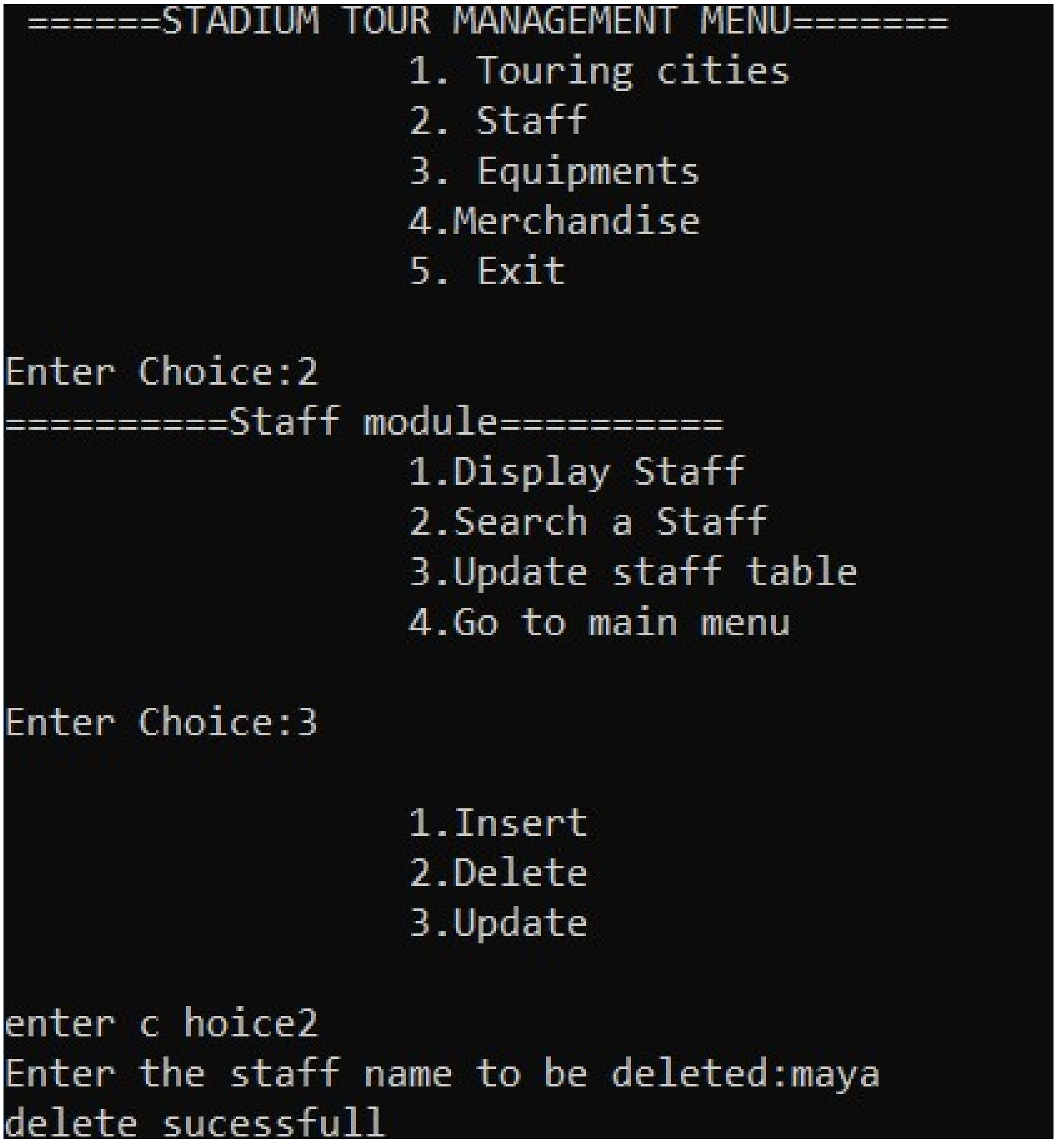
```
Enter Choice:2
==========Staff module==========
                  1.Display Staff
                  2.Search a Staff
                  3.Update staff table
                  4.Go to main menu

Enter Choice:3

                  1.Insert
                  2.Delete
                  3.Update

enter c hoice1
Enter stid:2002
Enter snamemaya
enter jobbackup dancer
One record added successfully.
==========Staff module==========
                  1.Display Staff
                  2.Search a Staff
                  3.Update staff table
                  4.Go to main menu

Enter Choice:1
Stid    Staff name      Job
1234     melissa         Background Singer
4591     stella          Background Singer
```

Updating in staff module



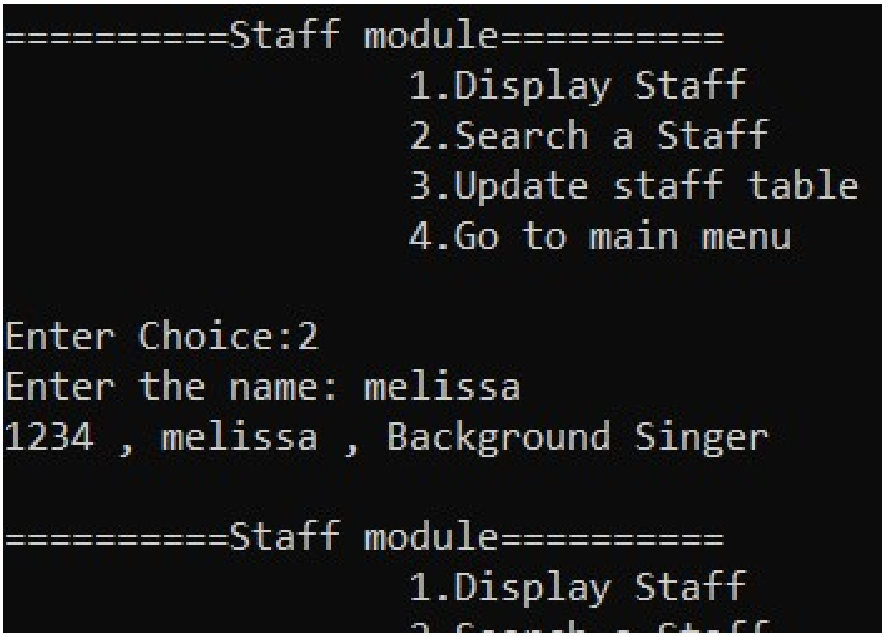
```
enter c hoice3
Enter the staff name to be updatemaya
Enter stid:1222
Enter snamemaia
enter jobbackup dancer
record updated sucessfully
==========Staff module==========
                  1.Display Staff
                  2.Search a Staff
                  3.Update staff table
                  4.Go to main menu

Enter Choice:1
Stid    Staff name      Job
1234     melissa         Background Singer
4591     stella          Background Singer
5609     bob     technician
8905     joel    guitarist
2002     maya    backup dancer
```

Deleting from staff module



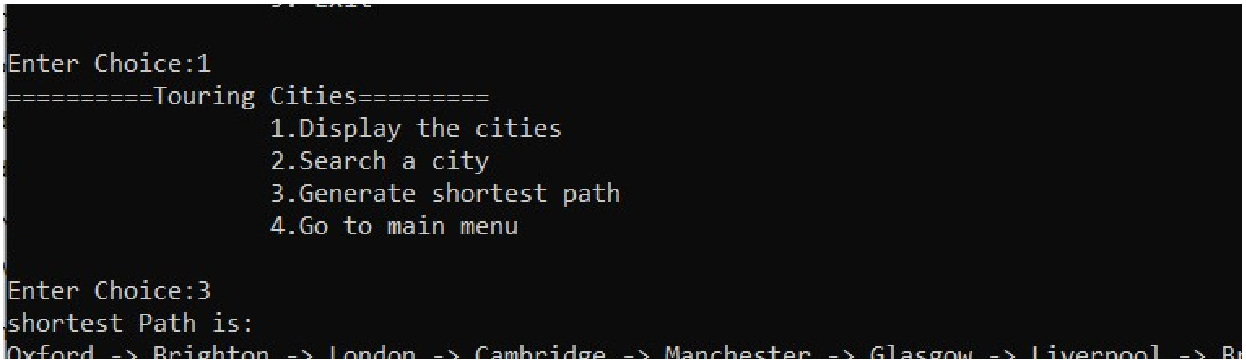
```
 ======STADIUM TOUR MANAGEMENT MENU=======
                  1. Touring cities
                  2. Staff
                  3. Equipments
                  4.Merchandise
                  5. Exit

Enter Choice:2
==========Staff module==========
                  1.Display Staff
                  2.Search a Staff
                  3.Update staff table
                  4.Go to main menu

Enter Choice:3

                  1.Insert
                  2.Delete
                  3.Update

enter c hoice2
Enter the staff name to be deleted:maya
delete sucessfull
```

Search operation in staff module-



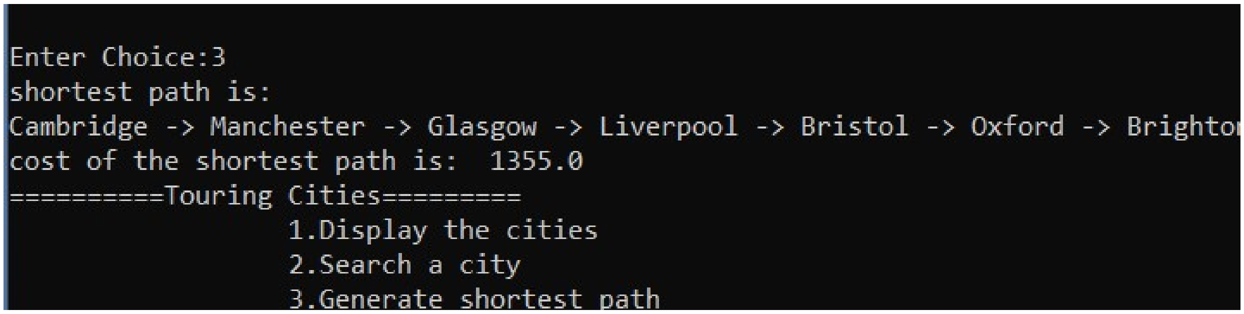
```
==========Staff module==========
                  1.Display Staff
                  2.Search a Staff
                  3.Update staff table
                  4.Go to main menu

Enter Choice:2
Enter the name: melissa
1234 , melissa , Background Singer

==========Staff module==========
                  1.Display Staff
                  2.Search a Staff
```

Displaying shortest path covering all cities using dynamic programming



```
Enter Choice:1
==========Touring Cities=========
                  1.Display the cities
                  2.Search a city
                  3.Generate shortest path
                  4.Go to main menu

Enter Choice:3
shortest Path is:
Oxford -> Brighton -> London -> Cambridge -> Manchester -> Glasgow -> Liverpool -> Br
```

Displaying shortest path covering all cities generated using genetic algorithm



```
Enter Choice:3
shortest path is:
Cambridge -> Manchester -> Glasgow -> Liverpool -> Bristol -> Oxford -> Brightor
cost of the shortest path is:  1355.0
==========Touring Cities=========
                  1.Display the cities
                  2.Search a city
                  3.Generate shortest path
```

CONCLUSION

In this project, we used to DP and GA contrast with create minimum spanning path. The calculations were applied of distance matrix of eight cities. In particular, we utilized diagrams to analyze the time taken and effectiveness of every calculation. The consequences of the analyses show the accompanying:

(1) The exhibitions of the two calculations are unequivocally influenced by their parameters just as their applications. GA is skilled at looking for ideal arrangements in low input cases. Now and again, for example, with various input greater than 100, the presentation of GA is better than that of DP. In any case, for lower amount of data GA takes additional time than DP.

(2) DP has a decent presentation in acquiring steady and top notch solutions, even in looking through huge search space. Be that as it may, because of its exhaustive enumerating nature, it costs abundance time in huge amount of data cases.

As of late, touring has become the significant wellspring of salary for music industry. Arena visit the executives framework makes the logistics part of touring simpler and increasingly practical. It will decrease any utilization of paper in keeping up information about visit.

This undertaking work is planned for making touring progressively conservative. Different tables, for example, stock, rundown of urban communities, supplies and so forth are incorporated. This project also contains a function for generating shortest path spanning all the cities. This will help in covering all cities in a much smaller total distance. Anyway this venture doesn't mull over a few things like the different levies forced while travelling interstate or the accessibility of arenas on required dates and so on. With everything taken into account whole undertaking may come convenient for touring chiefs.

.

CITATIONS

- Project idea Source- JeffyLo94. (2017, November 20). JeffyLo94/CS1D-Stadium-Travel-Project. Retrieved from https://github.com/JeffyLo94/CS1D-Stadium-Travel-Project.

- Owens, M. (2014). Definitive guide to sqlite. Springer-verlag Berlin And Hei.

- Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2006). The traveling salesman problem: a computational study.

- Al-Dallal, A. (2015). Using Genetic Algorithm with Combinational Crossover to Solve Travelling Salesman Problem. Proceedings of the 7th International Joint Conference on Computational Intelligence. doi: 10.5220/0005590201490156

- Gupta, S. B. (2010). Introduction to database management systems. Place of publication not identified: Laxmi Publications.

- Wei, J.-D. (2008). Approaches to the Travelling Salesman Problem Using Evolutionary Computing Algorithms. Traveling Salesman Problem. doi: 10.5772/5584

- Stoltz, E. (2018, July 17). Retrieved from https://towardsdatascience.com/evolution-of-a-salesman-a-complete-genetic-algorithm-tutorial-for-python-6fe5d2b3ca35