

**JAYPEE INSTITUTE OF INFORMATION  
TECHNOLOGY  
SECTOR 62, NOIDA  
UTTAR PRADESH**



**ALGORITHMS AND PROBLEM SOLVING  
PROJECT REPORT**

**CITY TOUR NAVIGATION OPTIMIZATION SYSTEM**

**BATCH – B11**

**SUBMITTED TO : Dr. DHANALEKSHMI G**

**SUBMITTED BY :**

KAMAL GARG	21103231
DIVYA	21103233
SRISHTI GARG	21103227
HIMANSHU DIXIT	21103262

## **ACKNOWLEDGEMENT**

We wish to express our profound and deep sense of gratitude to Prof. Dhanalekshmi, Prof Anita Sahoo, Prof. Ankita Wadhwa for motivating us to make such a report and for sparing time to go through every tiny detail and give his valuable suggestions to make this project report a success. We are mainly indebted to the authors of many references and articles which were used as the reference. Last but not the least we would like to thank our friends and family for their help in every way for the success of this project report.

## **INTRODUCTION**

On seeing the current need we thought of a system that can guide people at unknown places to reach their destination in an easy and optimized way.

we Have designed our Map guide system with functionalities that can help people reach their destination through shortest paths which is time efficient and financially efficient. We have Transport guidance functionality which helps you to choose efficient options available. We also have Tourist guidance functionality which can help you tour a place with optimized and efficient paths.

## **BACKGROUND**

Why is it essential to have a guidance system at unknown places?

I think we can agree that when traveling, we would want to experience the place as best and as much as possible. One way to help you with that is having the guidance system for the we are visiting.

Traveling is fun, but can sometimes be challenging. Many countries are very different from the states. That's why it can be more difficult and challenging if you are traveling abroad because things are so different and you are not used to them. With the language, culture and system differences, it can be really confusing and overwhelming, especially if you've never been before.

It's fine if you want to visit the generic major tourist destinations with big crowds, but don't you want to experience the off-the-beaten-track spots or have more local experience?

That why is becomes important to have a guidance system for unknown places, as it not only helps you

To find different location easily but it also saves your time and cost.

## **BENEFITS OF HAVING GUIDANCE SYSTEM**

### **1. Enhanced trip:**

Having a tour guidance system by your side during your trip can enhance your experience when traveling. As it helps you to focus more on visiting and enjoying your trip than to worry about the path to reach destination.

They can also bring you to off-the-beaten-path spots you wouldn't have known otherwise and offer personal suggestions on how to get the most out of your trip.

### **2. It's hassle-free:**

One of the things that may be holding you back from having a great trip is that you have to figure out the small details when traveling around.

Where to go? How to get around? Directions? What time does the place open or close? If you use a guidance system, then you don't have to trouble yourself with it.

### **3. Language barrier:**

Another thing is the language difference. It's difficult to communicate especially if you are traveling to places where the people speak little to no English at all. But if you have guidance system do not have to talk people to ask path for visiting different places

#### 4. You can uncover lesser known places:

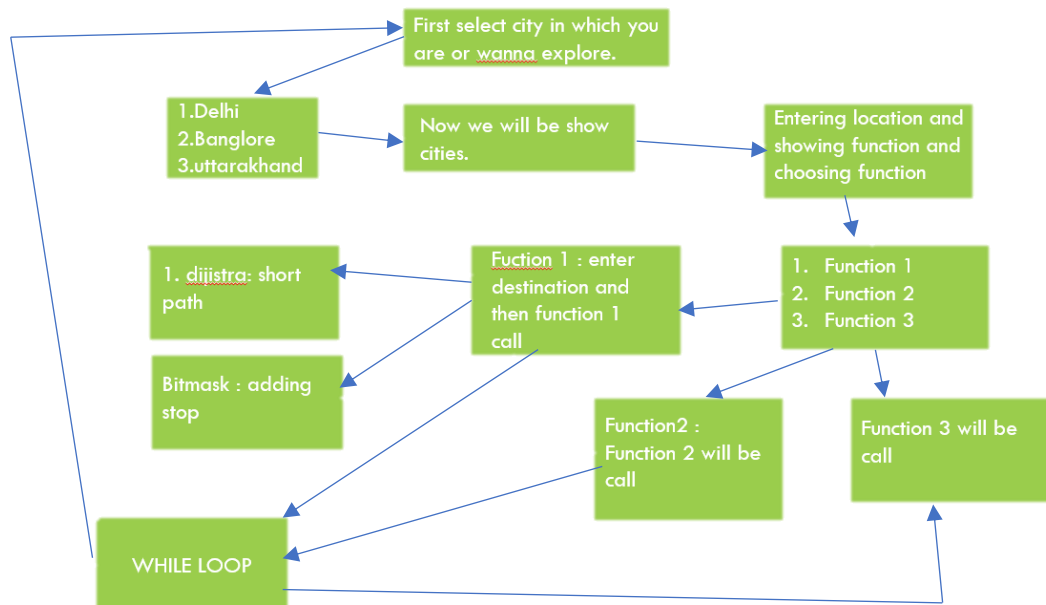
You can visit off-the-beaten-track spots and a new place that is out of the major tourist spots with the help of a guide. Don't you like the thrill of finding places beyond the obvious and aren't on popular tourist routes? You can be amazed how the off-the-beaten-path spots can be equally and even more incredible than the major ones.

#### 5. It's time and cost efficient:

You don't have to waste your time to plan a good travel itinerary, as the guides will do it for you. A good guidance system will structure your itinerary to be time-efficient, probably allowing you to visit more places.

You are probably thinking that how guidance system can it save money? Here's how. You don't have to figure out the route by yourself and risking getting lost which can cost you more money and also it saves your money since you do not have to hire guides for planning your journey.

## FLOW CHART



## DATA STRUCTURES USED

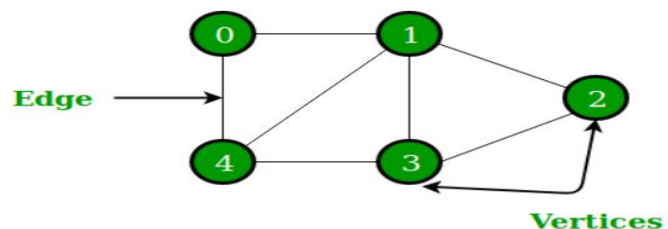
### 1. Priority queue / Heap:

A heap is a specific tree based data structure in which all the nodes of tree are in a specific order. Let's say if X is a parent node of Y, then the value of X follows some specific order with respect to value of Y and the same order will be followed across the tree. The maximum number of children of a node in the heap depends on the type of heap. However in the more commonly used heap type, there are at most 2 children of a node and it's known as a Binary heap. In binary heap, if the heap is a complete binary tree with N nodes, then it has the smallest possible height which is  $\log_2 N$ .

## 2. Graphs:

A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph.

In the Graph shown, the set of vertices  $V = \{0,1,2,3,4\}$  and the set of edges  $E = \{01, 12, 23, 34, 04, 14, 13\}$ .



## 3. Hashmap / Hashtable:

Hash Table is a data structure which stores data in an associative manner. In a hash table, data is stored in an array format, where each data value has its own unique index value. Access of data becomes very fast if we know the index of the desired data. Thus, it becomes a data structure in which insertion and search operations are very fast irrespective of the size of the data. Hash Table uses an array as a storage medium and uses hash technique to generate an index where an element is to be inserted or is to be located from.

## 4. Array:

Arrays are a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.



## **ALGORITHMS USED**

### **1. Dijkstra's Algorithm:**

Dijkstra's algorithm is also known as the shortest path algorithm. It is an algorithm used to find the shortest path between nodes of the graph. The algorithm creates the tree of the shortest paths from the starting source vertex from all other points in the graph. It differs from the minimum spanning tree as the shortest distance between two vertices may not be included in all the vertices of the graph. The algorithm works by building a set of nodes that have a minimum distance from the source. Here, Dijkstra's algorithm uses a greedy approach to solve the problem and find the best solution.

### **2. Prim's algorithm:**

Prim's algorithm is also a Greedy algorithm. It starts with an empty spanning tree. The idea is to maintain two sets of vertices. The first set contains the vertices already included in the MST, the other set contains the vertices not yet included. At every step, it considers all the edges that connect the two sets, and picks the minimum weight edge from these edges. After picking the edge, it moves the other endpoint of the edge to the set containing MST. A group of edges that connects two set of vertices in a graph is called cut in graph theory. So, at every step of Prim's algorithm, we find a cut, pick the minimum weight edge from the cut and include this vertex to MST Set.

### 3. Kruksal's algorithm:

Kruskal's Algorithm is used to find the minimum spanning tree for a connected weighted graph. The main target of the algorithm is to find the subset of edges by using which, we can traverse every vertex of the graph.

## **PROJECT DESCRIPTION**

### BRIEF OF THE PROJECT

We have made a user friendly Map guide system. It is easy to operate and features are accessible to all the users who are new to the city or living in the selected city and want to travfrom one place to another Or want to visit all places in the current city .

We have used different algorithms to optimize every feature of this map guide system such as Dijkstra Algorithm, Prims algorithms and Kruskal algorithm so that every feature will take minimum run time.

We have stored maps of different cities in form of undirected graphs in our database so that user is able to choose among the city maps or choices of cities.

Here are some of the features of our project .

- PROVIDING ALL PATHS FROM ONE SOURCE TO DESTINATION

After selecting city from the provided options we ask user to give its current location and destination that he/she is required to visit. Then

after getting that information from current user and analysing our database we will provide information regarding all the paths that are available from user's source to destination. This feature is implemented using depth first search.

- PROVIDING SHORTEST PATH FROM SOURCE TO DESTINATION

In this feature If user is willing to go through the shortest path then he/she can convey his/her willingness then we will show the shortest path from that source to destination. If there more than one then we will show all the shortest paths.

This feature is implemented using Dijkstra algorithm

- PROVIDING A WAY IF USER WANTS TO VISIT ALL THE PLACES IN MAP

If user is new to current city and he/she wants to visit All the places in the current city then we will Show him/her shortest path so that the user can visit all the places In the shortest amount of time. This feature is implemented using prims/Kruskal algorithm.

- PROVIDING OPTION TO USER TO ADD MULTIPLE STOPS IN BETWEEN SOURCE AND DESTINATION

As we already witness in ola and uber apps we can add multiple stops in between our source and destination . In this project we will also give user a option to add multiple stops in between source and destination and after processing user's choices we will gave shortest path that will must cover those stops and lead to destination. This feature is implemented using bit masking and DFS algorithm.

- PROVIDING TRANSPORT GUIDANCE TO THE USER

In this feature if we ask the user if user has his own vehicle if user does not own vehicle then we will show current available drivers at user's current location and

Show details of cars that the driver own and their asking price per km and other necessary details so that user can choose the driver that is most suitable for him/her. This feature is implemented using hash maps

## OUTPUT SCREENSHOTS:

[illegible]

HERE ARE THE MAPS OF THE CITIES THAT ARE AVAILABLE IN OUR DATA BASE

1. DELHI
2. BANGALORE
3. UTTARAKHAND

Choose the city among about in which you live/want to travel :  
DELHI

\*\*\*\*\*Here is the list of places that you can visit in delhi\*\*\*\*\*

- 1.CHANDNICHOWK
- 2.MANDOLI
- 3.AKSHARDHAM
- 4.JANTARMANTAR
- 5.BHAJANPURA
- 6.INDIAGATE
- 7.REDFORT
- 8.KHAZORIKHAS
- 9.DILSHADGARDEN
- 10.OLDSEEMAPURI
- 11.CANNAUGHTPLACE
- 12.JANAKPURI

Enter your location :  
REDFORT

Now we know which city you are residing or wanna explore now choose one of the option from given options  
1. Want to travel from your locaion to some other location  
2. Want to visit all the places of current city in our database  
3. Want a transport guidance at your current location

(Press 1/2/3) to proceed

```
121 :- redfort->->indiagate->cannaughtplace->akshardham->bhajanpura->mandoli 33.9
122 :- redfort->->indiagate->cannaughtplace->akshardham->bhajanpura->dilshadgarden->mandoli 39.8
123 :- redfort->->indiagate->cannaughtplace->akshardham->khazoorikhas->bhajanpura->mandoli 38.3
124 :- redfort->->indiagate->cannaughtplace->akshardham->khazoorikhas->bhajanpura->dilshadgarden->mandoli 44.2
125 :- redfort->->indiagate->cannaughtplace->akshardham->khazoorikhas->mandoli 44.2
126 :- redfort->->indiagate->cannaughtplace->jantarmantar->janakpuri->dilshadgarden->bhajanpura->khazoorikhas->mandoli 80.6
127 :- redfort->->indiagate->cannaughtplace->jantarmantar->janakpuri->dilshadgarden->bhajanpura->akshardham->khazoorikhas->mandoli 104.2
128 :- redfort->->indiagate->cannaughtplace->jantarmantar->janakpuri->dilshadgarden->bhajanpura->mandoli 69.9
129 :- redfort->->indiagate->cannaughtplace->jantarmantar->janakpuri->dilshadgarden->mandoli 60.8
130 :- redfort->->indiagate->cannaughtplace->jantarmantar->chandnichowk->janakpuri->dilshadgarden->bhajanpura->khazoorikhas->mandoli 87.8
131 :- redfort->->indiagate->cannaughtplace->jantarmantar->chandnichowk->janakpuri->dilshadgarden->bhajanpura->akshardham->khazoorikhas->mandoli 111.4
132 :- redfort->->indiagate->cannaughtplace->jantarmantar->chandnichowk->janakpuri->dilshadgarden->bhajanpura->mandoli 77.1
133 :- redfort->->indiagate->cannaughtplace->jantarmantar->chandnichowk->janakpuri->dilshadgarden->mandoli 68
134 :- redfort->->indiagate->cannaughtplace->janakpuri->dilshadgarden->bhajanpura->khazoorikhas->mandoli 78.4
135 :- redfort->->indiagate->cannaughtplace->janakpuri->dilshadgarden->bhajanpura->akshardham->khazoorikhas->mandoli 102
136 :- redfort->->indiagate->cannaughtplace->janakpuri->dilshadgarden->bhajanpura->mandoli 67.7
137 :- redfort->->indiagate->cannaughtplace->janakpuri->dilshadgarden->mandoli 58.6
```

Do you want to see shortest path among these paths from redfort to mandoli (Press 0/1)

1

\*\*\*\*\*HERE ARE THE SHORTEST PATHS\*\*\*\*\*

Path 1:- redfort->mandoli - 14 KM

Do you want to enter stops in between you location -: redfort and destination :- mandoli (Press 0/1)

1

Now enter your stops that you want between your source and destination :  
janakpuri  
akshardham

Here are the paths from source to destination including stops in increasing order of distance

path 1 -: mandoli-> -> dilshadgarden->janakpuri->cannaughtplace->akshardham->indiagate->redfort distance-: 72  
path 2 -: mandoli-> -> dilshadgarden->janakpuri->jantarmantar->cannaughtplace->akshardham->indiagate->redfort distance-: 74.2  
path 3 -: mandoli-> -> bhajanpura->dilshadgarden->janakpuri->cannaughtplace->akshardham->indiagate->redfort distance-: 81.1  
path 4 -: mandoli-> -> dilshadgarden->janakpuri->chandnichowk->jantarmantar->cannaughtplace->akshardham->indiagate->redfort distance-: 81.4  
path 5 -: mandoli-> -> bhajanpura->akshardham->cannaughtplace->janakpuri->dilshadgarden->redfort distance-: 82.4  
path 6 -: mandoli-> -> bhajanpura->dilshadgarden->janakpuri->jantarmantar->cannaughtplace->akshardham->indiagate->redfort distance-: 83.3  
path 7 -: mandoli-> -> bhajanpura->akshardham->cannaughtplace->jantarmantar->janakpuri->dilshadgarden->redfort distance-: 84.6  
path 8 -: mandoli-> -> bhajanpura->akshardham->indiagate->cannaughtplace->janakpuri->dilshadgarden->redfort distance-: 86.6  
path 9 -: mandoli-> -> bhajanpura->khazoorikhas->akshardham->cannaughtplace->janakpuri->dilshadgarden->redfort distance-: 86.8  
path 10 -: mandoli-> -> bhajanpura->akshardham->indiagate->cannaughtplace->jantarmantar->janakpuri->dilshadgarden->redfort distance-: 88.8  
path 11 -: mandoli-> -> bhajanpura->khazoorikhas->akshardham->cannaughtplace->jantarmantar->janakpuri->dilshadgarden->redfort distance-: 89  
path 12 -: mandoli-> -> bhajanpura->dilshadgarden->janakpuri->chandnichowk->jantarmantar->cannaughtplace->akshardham->indiagate->redfort distance-: 90.5

Now we know which city you are residing or wanna explore now choose one of the option from given options

1. Want to travel from your locaion to some other location
2. Want to visit all the places of current city in our database
3. Want a transport guidance at your current location

(Press 1/2/3) to proceed

2

Here is the shortest path visiting all the places in the given city at minimum cost :

redfort{ 0 }-> indiagate{ 6.4 }-> cannaughtplace{ 3.1 }-> jantarmantar{ 2.2 }-> chandnichowk{ 4.2 }-> akshardham{ 7.7 }-> dilshadgarden{ 12 }-> mandoli{ 3.1 }-> bhajanpura{ 4.7 }-> khazoorikhas{ 2.4 }-> oldseemapuri{ 15 }-> janakpuri{ 17 }->

TOTAL COST OF VISITING IS : 77.8 KM

Do you want to visit your city menu again (Press 0/1)

Enter your location :

mandoli

Now we know which city you are residing or wanna explore now choose one of the option from given options

1. Want to travel from your locaion to some other location
2. Want to visit all the places of current city in our database
3. Want a transport guidance at your current location

(Press 1/2/3) to proceed

3

So You Need Transport Guidance Dont Worry We Have Driver Information in our Database  
Here Is The Information Regarding The Drivers Available At Your Location

1

Driver name -Monty  
Driver car name - Mahindra TUV 300  
Driver car no plate - DL5CH87671  
Driver Phone number - 9954992345  
Driver charges for per km 14 per km  
Please Contact These Drivers

Do you want to visit your city menu again (Press 0/1)

## **TIME COMPLEXITY ANALYSIS**

1.Our first feature is print all paths from source to destination which we have done by using dfs.

Its time complexity is  $O(V+E)$  where  $V$  is the no of vertices and  $E$  is no of edges in our graph

And its space complexity is  $O(V)$ .

2.Our second feature is to find shortest path by using Dijkstra's Algorithm. The time complexity remains  $O(E \log V)$  as there will be at most  $O(E)$  vertices in priority queue and  $O(\log E)$  is same as  $O(\log V)$ .

3.Our third feature is to allow users to add stops between source and destination .it is implemented using dfs and bitmasking. Its time complexity is  $O(V+E)$  where  $V$  is the no of vertices and  $E$  is no of edges in our graph

And its space complexity is  $O(V)$ .

4.Our fourth feature is to visit each and every place in map with minimum cost . It is implemented using prims algorithm. The time complexity of the Prim's Algorithm is  $O((V + E) \log V)$  because each vertex is inserted in the priority queue only once and insertion in priority queue take logarithmic time.

5.Our last feature is for driver assistance at particular location .it is implemented using unordered maps.its time complexity is  $O(N)$  where  $N$  is no of drivers at that location.

.

## **FUTURE WORK**

However, there is further work that can be done.

1. One unimplemented functions like share tour plan with others can be implemented.
2. We can add a reminder component, in which user can add their plans of travelling according to which it automatically reminds you, where you have to go next.
3. We can add a time component by which user can get a idea of how much time he/she needed to reach his/her destination.
4. We also add the feature of Weather in which we can give the idea of user's current location and destination weather information.
5. Feature like Street view, in which we shows pictures and images of the location. Which is definitely helpful to our users to make any decision regarding that location.



## **CONCLUSION**

In conclusion, it is not easy to complete this project without the coordination and communication among the team members involved in the project. The purpose of this project was to create a system that can guide people to unknown places and to help them reach their destination in an easy and optimized way. Based on our work, we have created multiple options for our users such as getting the shortest path or multiple other options related to path as per their requirements along with the transport guidance system for the user.