**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnana Sangama”, Belagavi-560018, Karnataka**

****

**DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY**

**WITH MINI PROJECT - 19IS4DLADA**

**REPORT**

**On**

**“PROFIT ORIENTED TRANSPORTATION”**

**BACHELOR OF ENGINEERING**

In

**INFORMATION SCIENCE AND ENGINEERING**

Submitted by

**Reinal Antonio Carlo (1DS19IS077)**

**Sai Ram V (1DS19IS082)**

**Sai Sumanth M (1DS19IS083)**

**Samarth BS (1DS19IS084)**

**LAB IN-CHARGE**

Mrs. Bindu Bhargavi S M

Mrs. Shilpa Shree S

Mrs. Sreevidya B S



**2020-21**

**Department of Information Science and Engineering**

**DAYANANDA SAGAR COLLEGE OF ENGINEERING**

**SHAVIGE MALLESHWARA HILLS, KUMARASWAMY LAYOUT**

**Bengaluru-560078**

**DAYANANDA SAGAR COLLEGE OF ENGINEERING**

**Shavige Malleshwara Hills, Kumaraswamy Layout-560078**

**Department of Information Science and Engineering**

****

**2020-21**

**CERTIFICATE**

This is to certify that Design and Analysis of Algorithms Laboratory with Mini Project Work (19IS4DLADA) entitled **PROFIT ORIENTED TRANSPORTATION.** Is a bonafide work carried out by **Students Reinal Antonio Carlo (1DS19IS077), Sai Ram V (1DS19IS082), Sai Sumanth M (1DS19IS083) and Samarth BS (1DS19IS084)** in the partial fulfillment for the 4th semester of **Bachelor of Engineering in Information Science and Engineering** of the Visvesvaraya Technological University, Belgavi during the year 2020-21.

*Signature of Lab-In charge*

Name of the Examiners Signature with Date

1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Topics** | **Page No** |
| **1** | Introduction | **4** |
|  | 1.1 Dynamic Programming | **4** |
|  | 1.2 Greedy Technique | **4** |
| **2** | System Requirements | **5** |
|  | 2.1 Hardware Requirements | **5** |
|  | 2.2 Software Requirements | **5** |
| **3** | Algorithms | **6-7** |
|  | 3.1 Floyd Warshall Algorithm | **6** |
|  | 3.1.1 Pseudocode | **6** |
|  | 3.2 Prim’s Algorithm | **7** |
|  | 3.2.1 Pseudocode | **7** |
| **4** | Implementation | **8-15** |
|  | 4.1 Implementation of Floyd Warshall Algorithm | **8-10** |
|  | 4.2 Implementation of Prim’s Algorithm | **11-13** |
|  | 4.3 Implementation of cost and time | **13** |
|  | 4.4 Snapshots of result | **14-15** |
| **5** | Time complexity | **16** |
|  | 5.1 Time Complexity of Floyd Warshall Algorithm | **16** |
|  | 5.1 Time Complexity of Prim’s Algorithm | **16** |
| **6** | Conclusion | **17** |
| **7** | Reference | **17** |

**1. Introduction**

Meeting the needs of middle class people and thinking what would help them when they plan to travel in most profitable way. In this project, we help user to find out the minimum distance they need to travel one place to another as well as minimum distance they need to travel all the places. To travel one place to another we use Floyd Warshall Algorithm which is example of dynamic programming. This is used to solve shortest path problem. To travel all the places we use Prims Algorithm which is example of greedy technique. This is used to solve minimum spanning tree problems.

**1.1 Dynamic Programming**

Dynamic Programming refers to simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner. A problem can be solved optimally by breaking it into sub-problems and then recursively finding the optimal solutions to the sub-problems. The Floyd-Warshall algorithm is an example of dynamic programming. It breaks the problem down into smaller sub problems, and then combines the answers to those sub problems to solve the big, initial problem. The shortest path problem is about finding a path between vertices in a graph such that the total sum of the weights of its constituent edges is minimized.

**1.2 Greedy Technique**

Greedy Technique always makes the choice that seems to be the best at that moment. This means that it makes a locally-optimal choice in the hope that this choice will lead to a globally-optimal solution. Prim's algorithm is a greedy algorithm that finds a minimum spanning tree for a weighted undirected graph. The minimum spanning tree is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.

We have merged both these problems in our project. User can find minimum distance they need to travel all the places as well as minimum distance they need to travel one place to another. We not only focus on shortest path, we also focus on how we can travel by spending fewer amount of time and money. We show user what is the minimum travel expense and minimum time to travel by multiple means of transport. In this way user can know, which is the most profitable transportation. This project demonstrate the application of prims and Floyd Warshall algorithm to design

**2. System Requirements**

**2.1 Hardware Requirements**

* Intel (or AMD equivalent) i5 or better processor, 7th generation or newer
* Minimum 1GHz CPU with four or more cores (Recommended 2GHz or more)
* Minimum 1 GB RAM (Recommended 4 GB or above)
* Minimum 32 GB hard drive Recommended 64 GB or more

**2.2 Software Requirements**

* Code editor where you write the source code of your program
* Integrated Development Environment (IDE) to edit and compile the c program. The best IDE’s for c programming are visual studio code, Code::Blocks, Dev-C++, Geany.
* Windows or any compatible Operating system

**3. Algorithms**

The algorithms used in our projects are Floyd Warshall Algorithm and Prism’s Algorithm. Floyd Warshall Algorithm is used to find minimum distance to travel from one place to another. Prims Algorithm is used to find minimum distance to travel all the places in a given graph.

**3.1 Floyd Warshall Algorithm**

Prim's algorithm to find minimum cost spanning tree uses the greedy approach. Prim's algorithm shares a similarity with the shortest path first algorithms. Prim's algorithm finds the subset of edges that includes every vertex of the graph such that the sum of the weights of the edges can be minimized. Prim's algorithm starts with the single node and explores all the adjacent nodes with all the connecting edges at every step. The edges with the minimal weights causing no cycles in the graph got selected

For a graph with N vertices:

* Initialize the shortest paths between any 2 vertices with Infinity.
* Find all pair shortest paths that use 0 intermediate vertices, then find the shortest paths that use 1 intermediate vertex and so on, until using all N vertices as intermediate nodes.
* Minimize the shortest paths between any 2 pairs in the previous operation.
* For any 2 vertices (i,j) , one should actually minimize the distances between this pair using the first K nodes, so the shortest path will be: min(cost[i][k]+cost[k][j],cost[i][j]).

cost[k][j] represents the shortest path that only uses the first K vertices, cost[k][j] represents the shortest path between the pair k,j. As the shortest path will be a concatenation of the shortest path from i to k, then from k to j.

**3.1.1 Pseudocode**

for i = 1 to N

for j = 1 to N

if there is an edge from i to j

dist[0][i][j] = the length of the edge from i to j

else

dist[0][i][j] = INFINITY

for k = 1 to N

for i = 1 to N

for j = 1 to N

dist[k][i][j] = min(dist[k-1][i][j], dist[k-1][i][k] + dist[k-1][k][j])

**3.1 Prim’s Algorithm**

Prim's Algorithm is used to find the minimum spanning tree from a graph. It uses greedy technique. Prim's algorithm finds the subset of edges that includes every vertex of the graph such that the sum of the weights of the edges can be minimized. Given n points, connect them in the cheapest possible way so that there will be a path between every pair of points. This problem arises naturally in several practical situations. We can represent the points by vertices of a graph, possible connections by the graph's edges and the connection costs by the edge weights. Prim's algorithm starts with the single node and explores all the adjacent nodes with all the connecting edges at every step. The edges with the minimal weights causing no cycles in the graph got selected. Prim’s algorithm constructs a minimum spanning tree through a sequence of expanding subtrees. The initial subtree in such a sequence consists of a single vertex selected arbitrarily from the set V of the graph's vertices. On each iteration, we expand the current tree in the greedy manner by simply attaching to it the nearest vertex not in that tree.

The algorithm stops after all the graph's vertices have been included in the tree being constructed. Since the algorithm expands a tree by exactly one vertex on each of its iterations, the total number of such iterations is n - 1, where n is the number of vertices in the graph.

* Select a starting vertex
* Select an edge e connecting the tree vertex and fringe vertex that has minimum weight
* Repeat Step 2 until there are fringe vertices
* Add the selected edge and the vertex to the minimum spanning tree T
* Exit

**3.1.1 Pseudocode**

PrimsMST(G,w,r)

for each u ∈ G,V

u.key = ∞

u. π = NIL

r.key = 0

Q = G,V

while Q ≠ 0

u = EXTRACT-MIN(Q)

for each v ∈ G.Adj|u|

if v ∈ Q and w(u,v)<v.key

v. π = u

v.key = w(u,v)

**4. Implementation**

There are 2 parts in the program. Firstly finding the minimum distance and then finding the minimum cost and time. Our project helps both local people as well as a tourist. For local people, to find minimum distance to travel from one place to another, we use Floyd Warshall Algorithm. For tourists, to find minimum distance to travel all the places in a given graph, we use Prim’s Algorithm.

**4.1 Implementation of Floyd Warshall Algorithm**

Floyd-Warshall Algorithm is an algorithm for finding the shortest path between all the pairs of vertices in a weighted graph. This algorithm works for both the directed and undirected weighted graphs. Let’s take the example of the below graph

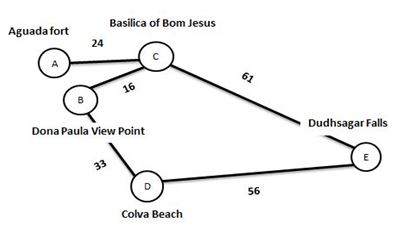


Figure 4.1. Example graph of Floyd Warshall

Let’s write initial distance matrix.

Each cell D[i][j] is filled with the distance from the ith vertex to the jth vertex. If there is no path from ith vertex to jth vertex, the cell is left as infinity.

D0=

Now, create a matrix D1 using matrix D0. The elements in the first column and the first row are left as they are. The remaining cells are filled in the following way.  
  
Let k be the intermediate vertex in the shortest path from source to destination. In this step, k is the first vertex. D[i][j] is filled with

(D[i][k] + D[k][j]) if (D[i][j] > D[i][k] + D[k][j])

If the direct distance from the source to the destination is greater than the path through the vertex k, then the cell is filled with D[i][k] + D[k][j].

D1 =

For example: For D0[3, 2], the direct distance from vertex 3 to 2 is 16 and the sum of the distance from vertex 3 to 2 through vertex (i.e. from vertex 3 to 1 and from vertex 1 to 2) is . Since 16 <D0 [3, 2] is filled with 16.

Similarly, D2 is created using D1. The elements in the second column and the second row are left as they are. In this step, k is the second vertex (i.e. vertex 2). The remaining steps are the same.

D2 =

Similarly D3, D4 and D5 is also created

D3 =

D4 =

D5 =

D5 gives the shortest path between each pair of vertices. We can obtain the minimum distance from one vertex to another from this all pair shortest path matrix. We ask user what is their source and destination place, and fetch the distance. If our source place is Aguada fort (i.e. vertex 1) and destination place is Dudhsagar falls (i.e. vertex 5), then our program fetches D5 [1, 5] = 85. So the minimum distance to travel from Aguada fort to Dudhsagar falls is 85kms.

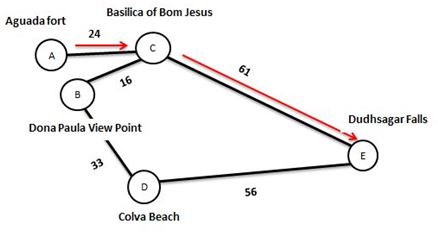


Figure 4.2. Visualization of Shortest path from Aguada fort to Dudhsagar Falls

**4.2 Implementation of Prim’s Algorithm**

Prim’s Algorithm is a famous greedy algorithm. It is used for finding the Minimum Spanning Tree (MST) of a given graph. Prim’s algorithm forms a tree that includes every vertex and has minimum sum of weights among all the tees that can be formed from the given graph. Let’s take example of same graph. We start from one vertex and keep adding edges with the lowest weight until we reach our goal.

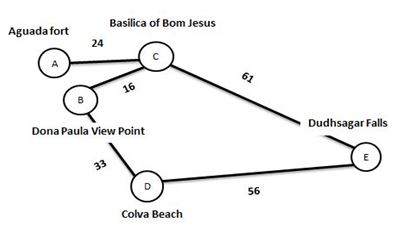


Figure 4.3 Example graph

Let’s initialize the minimum spanning tree with a vertex chosen at random. In our project, we ask user to select the places they want to travel. The first place is considered as the first vertex. Lets consider A as our first vertex.

Step: 2 choose the shortest edge from this vertex and add it

24

Step: 1 choose a vertex

C

A

A

B

C

A

16

24

Step: 3 Repeat Step 2 till you get spanning tree

Here, there are 2 adjacent vertexes for C (B and E). So we choose the shortest edge from C. i.e. B (16kms).

Since all the vertices have been included in the MST, so we stop.

33

16

24

E

D

B

C

A

Choose the nearest vertex not yet in the solution

A

33

16

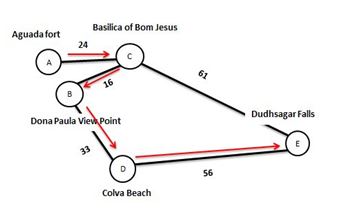
24

B

C

D

56



**Figure 4.4 Visualization of Shortest path to travel all the places in graph**

Now, Cost of Minimum Spanning Tree

= Sum of all edge weights

= 24 + 16 + 33 + 24

= 129 kms

So the minimum distance to travel all the places in graph is 129kms

**4.3 Implementation of cost and time**

Finding minimum cost and time required is very easy if we know the minimum distance, speed and cost per km. we know the minimum distance to travel from Floyd Warshall and Prims Algorithm. We have taken the average speed of bus and cab as 30 and 60km/hr. And cost per km of bus and cab as 1.5 and 9.5rs/km respectively. Multiplication of the distance and speed gives us time of travel in minutes. Multiplication of distance and cost per km gives us cost of travel. These results are not exact. It is just an approximation to give an idea about the travel

**4.4 Snapshots of results**

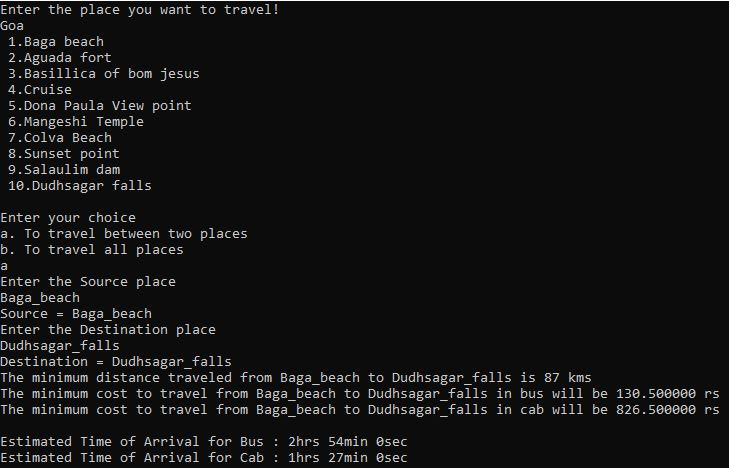
****

Figure 4.5. Output of the Floyd Warshall Algorithm. It gives minimum distance to travel from Baga beach to Dudhsagar falls. It also gives minimum travel expense and estimated time of travel.

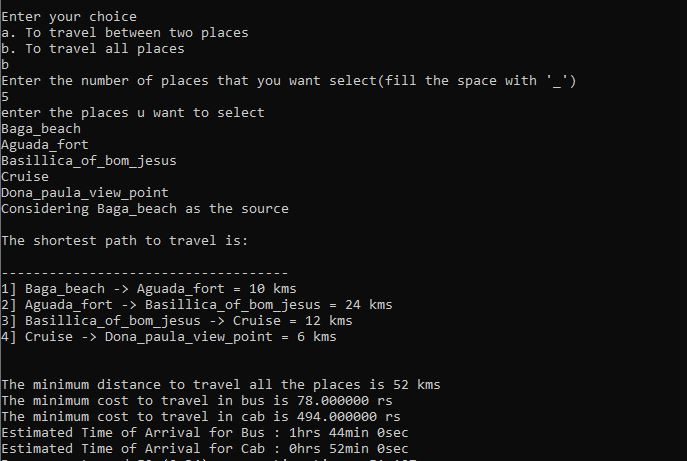


Figure 4.6. Output of the Prim’s Algorithm. It gives shortest path and minimum distance to travel from Baga beach to Dona paula view point. It also gives minimum travel expense and estimated time of travel.

**5. Time Complexity**

**5.1 Time Complexity of Floyd Warshall Algorithm**

Floyd Warshall Algorithm consists of three loops over all the nodes. The inner most loop consists of only constant complexity operations. Its complexity depends only on the number of vertices in the given graph. Hence, the asymptotic complexity of Floyd Warshall algorithm is **O(n3)**. Here, n is the number of nodes in the given graph. Its worst case, best case and average case complexity is **O(n3)**. Space complexity is **O(n2)**.

Worst case time complexity: **O(n3)**

Average case time complexity: **O(n3)**

Best case time complexity: **O(n3)**

Space complexity: **O(n2)**

**5.2 Time Complexity of Prim’s Algorithm**

Floyd Warshall Algorithm consists of two loops over all the nodes. The inner most loop consists of only constant complexity operations. Its complexity depends only on the number of vertices in the given graph. Hence, the asymptotic complexity of Floyd Warshall algorithm is O(n2). Here, n is the number of nodes in the given graph.

Worst case time complexity: **O(n2)**.

Average case time complexity: **O(n2)**.

Best case time complexity: **O(n2)**.

Space complexity: **O(E+V)**

**6. Conclusion**

This project is all about profit oriented transportation where user can find out which is the best possible way to travel all the places as well as one place to another. They can find out what is minimum distance to travel, which is cheapest way to travel, which is fastest way to travel. In this course of work it was discovered that the algorithm is very effective in providing the shortest distance between places. It helps reduces the transportation expense and time for transportation. This is helpful for tourists who are new to the places, they can figure out what is the estimated travel expense and time to reach their destination. This not only help tourist, it also helps the local travellers. This can be used for effective transportation of goods and services also. This method can also be explored by any organization, business, startups willing to achieve their goals with a reduced cost and effective use of their resources.

**7. Reference**

* “Introduction to the Design and Analysis of Algorithm” by Anany Levitin
* <https://www.gatevidyalay.com/prims-algorithm-prim-algorithm-example>/
* <https://www.programiz.com/dsa/prim-algorithm>
* <https://www.gatevidyalay.com/floyd-warshall-algorithm-shortest-path-algorithm>/
* <https://www.researchgate.net/publication/291907607_On_the_application_of_prim's_algorithm_to_a_profit-oriented_transportation_system_in_rural_areas_of_nigeria>
* <https://www.geeksforgeeks.org/finding-shortest-path-between-any-two-nodes-using-floyd-warshall-algorithm/>
* <https://www.programiz.com/dsa/floyd-warshall-algorithm>