BRACT'S
VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, KONDHWA(BK), PUNE-48 DEPARTMENT OF COMPUTER ENGINEERING

VISHWAKARMA INSTITUTES VI

Google Maps Implementation Using Dijkstra's Algorithm

Group No :- 11

Advanced Data Structure
SCE Presentation Year 2020-21

MEMBERS

74	Rohit Kudale
78	Shekhar Chaughule
79	Devesh Pathak
81	Masood Ismail Tamboli

CONTENTS

01 *Introduction to Project*

02 *What is Dijiktras Algorithm?*

03 *Working of Algorithm*

04 *Data Structures used*

05 *Pros and Cons*

06 *Resources used*

07 *Implementaion*

08 *Snapshots*

09 *Time and Space Complexity*

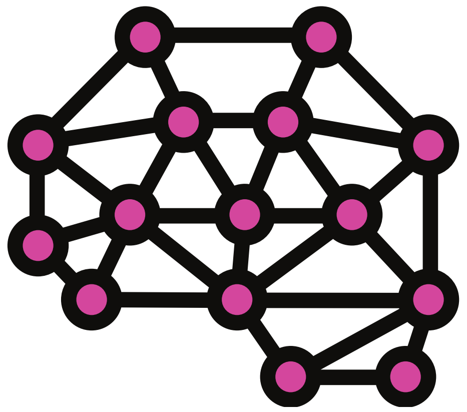
10 *Future Scope*

INTRODUCTION

- The goal of this project was to develop a basic prototype of Google Maps that finds the shortest path between source and destination.
- Being motivated by the fact that Google Maps uses the Dijktra's Algorithm to figure out the best route between two nodes, we attempted to build our simple Map of Maharashtra which contains apprx 18-20 cities with the real time distance between cities as cost/weights for the edges.
- Our project uses data such as Cities of Maharashtra and National Highways
- We created a Graph to show connections between Cities with Edges as National Highways and Distance as weights.
- We created a UI Design for it using PyQt5 and plotted the Graph of Cities as Nodes and Distance as Edges.

WHAT IS DIJIKTRAS ALGORITHM

- Dijkstra's Algorithm, an algorithm which is best suited for finding shortest path between nodes in a graph.
- This algorithm was created and published by Dr. Edsger Dijkstra
- A brilliant computer scientist and software engineer.
- In 1959, he published a 3-page article titled "A note on two problems in connection with graphs" where he explained his new algorithm.

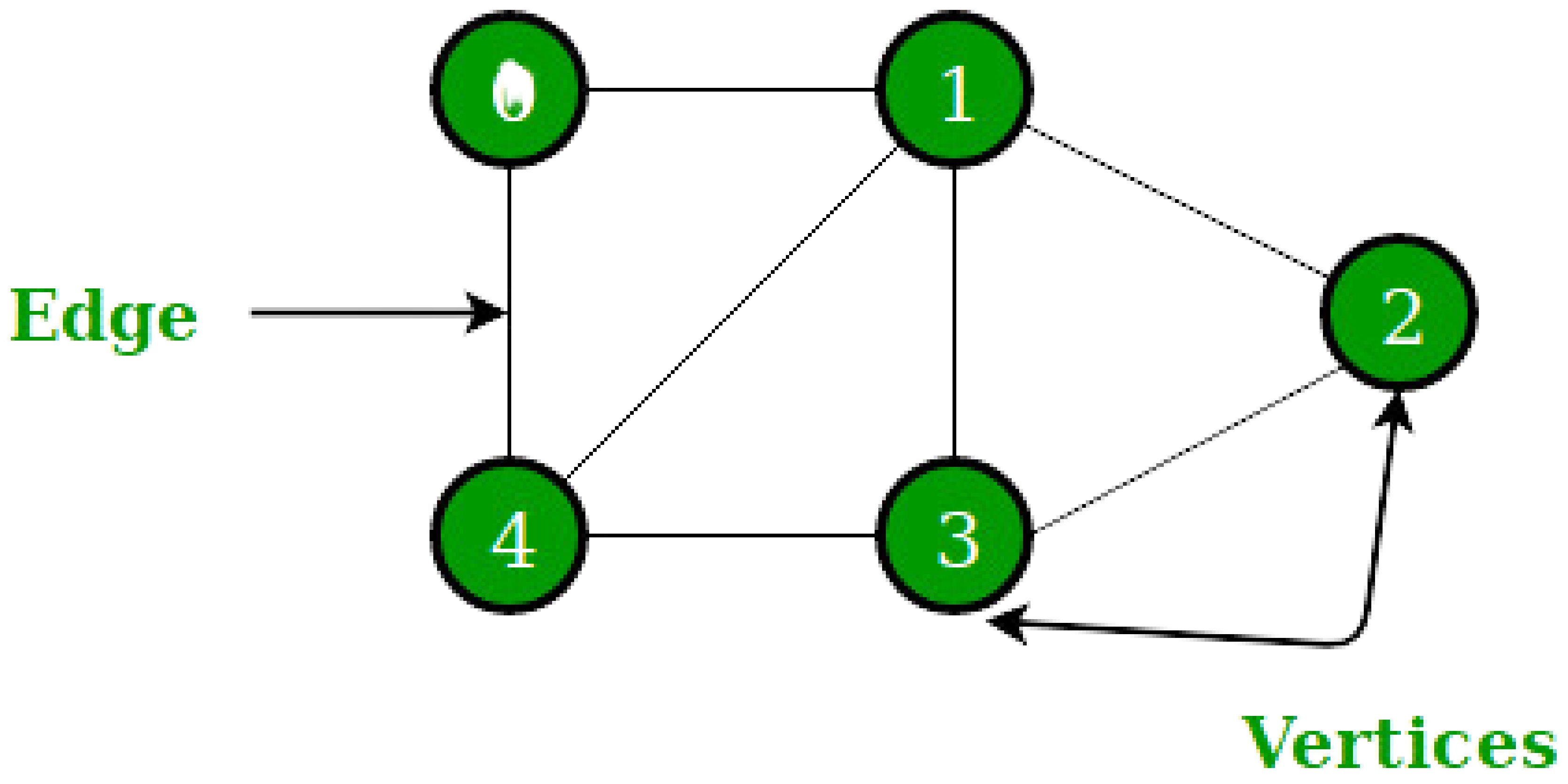


HOW THIS ALGO WORKS

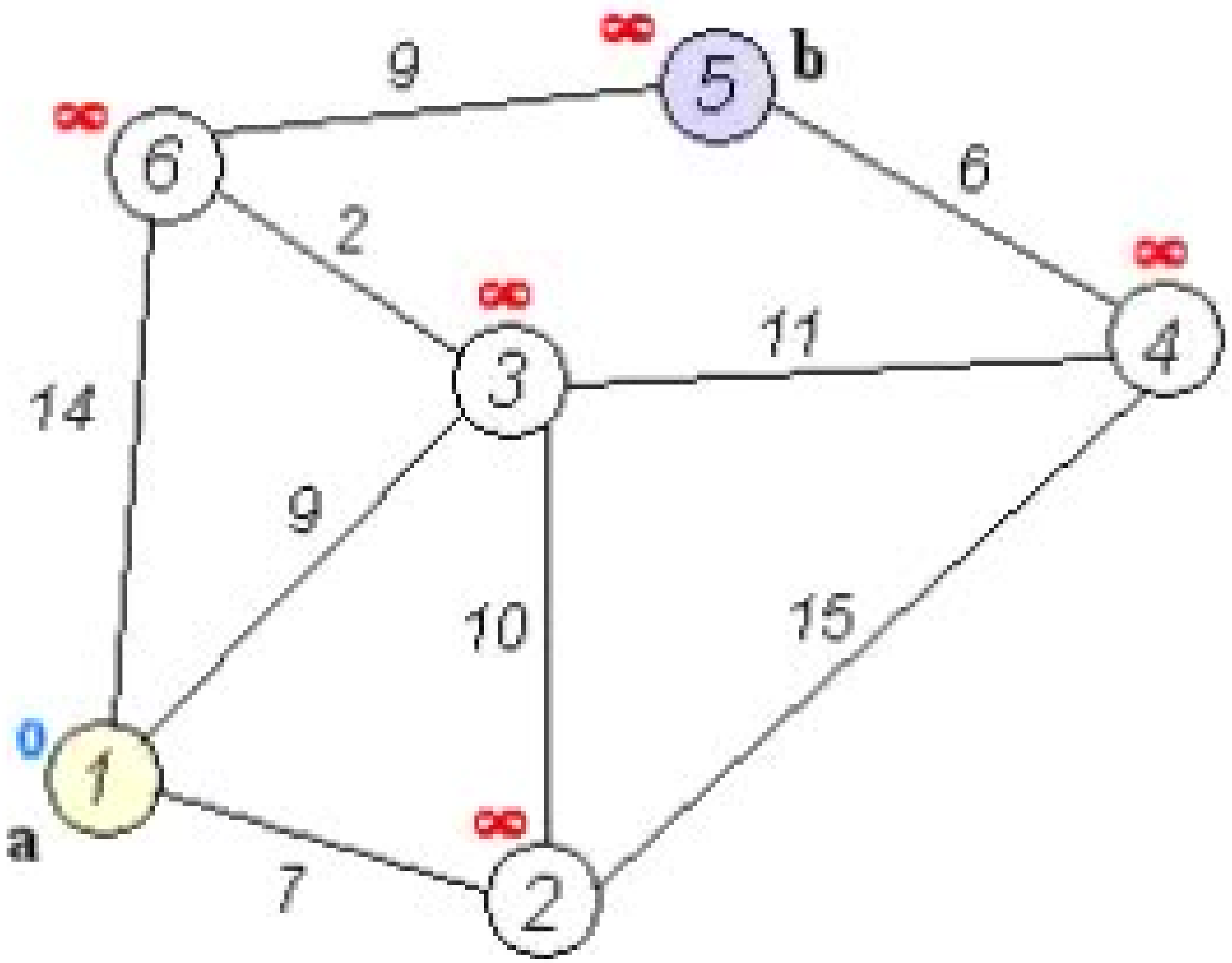
- Initialize distances according to the algorithm.
- Pick first node and calculate distances to adjacent nodes
- Pick next node with minimum distance;
- repeat adjacent node distance calculations.
- Final result of shortest-path from graph.

DATA STRUCTURES USED

For this project we have used graph data structure for building transportation systems, where intersection of two(or more) roads are considered to be a vertex and the road connecting two vertices is considered to be an edge, thus their navigation system is based on the algorithm to calculate the shortest path between two vertices.



1. Dijkstra's algorithm to find the shortest path between a and b. It picks the unvisited vertex with the lowest distance, calculates the distance through it to each unvisited neighbor(cities), and updates the neighbor's(cities) distance if smaller. Mark visited (set to red) when done with neighbors.



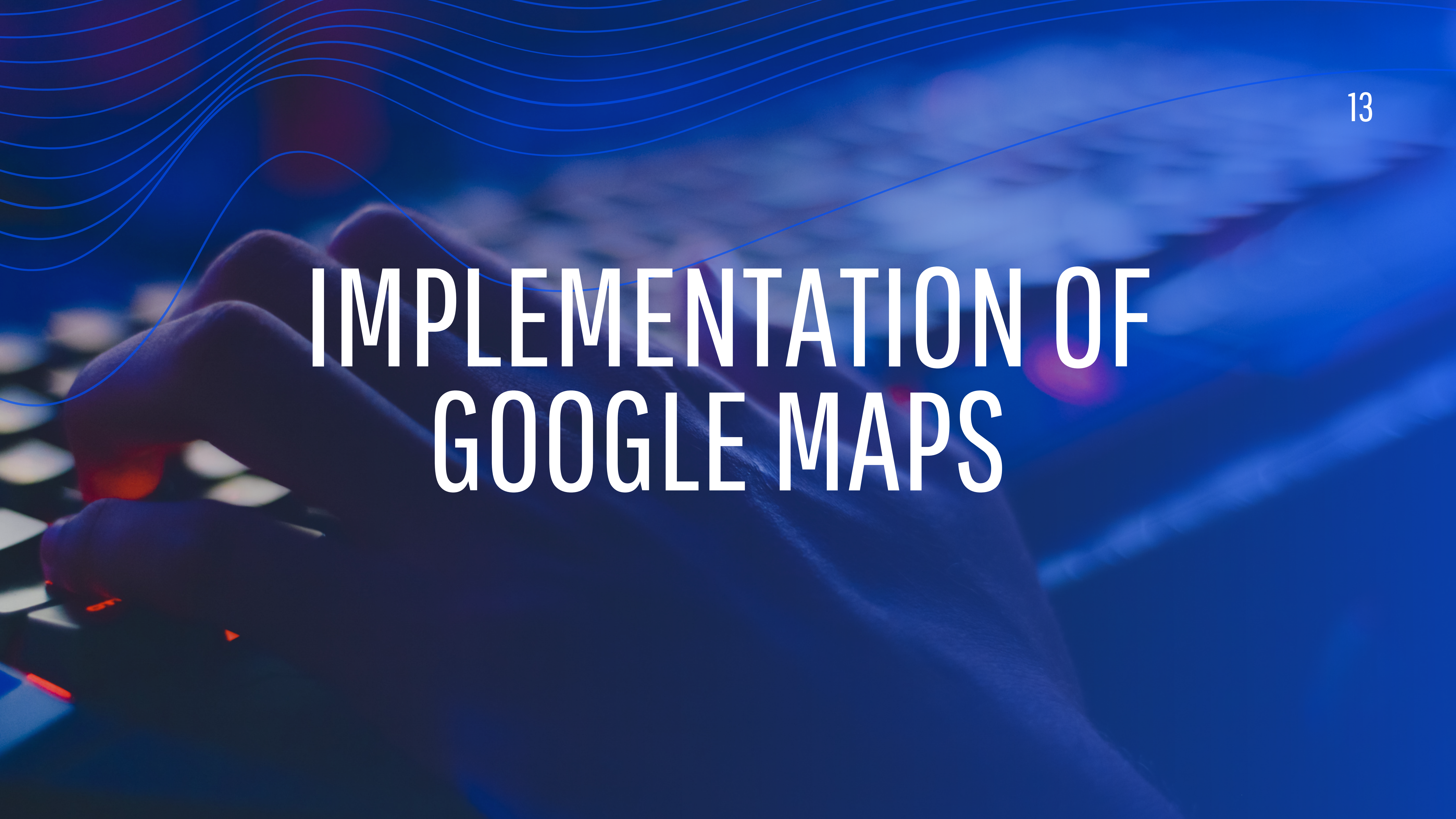
PROS AND CONS OF USING DIJIKTRAS ALGORITHM

PROS

- Helpful in finding Shortest Path.
- Easy to implement
- It can be used to calculate the shortest path between a single node to all other nodes and a single source node to a single destination node by stopping the algorithm once the shortest distance is achieved for the destination node
- It can work on Graphs i.e Directed graphs, weighted graphs etc.

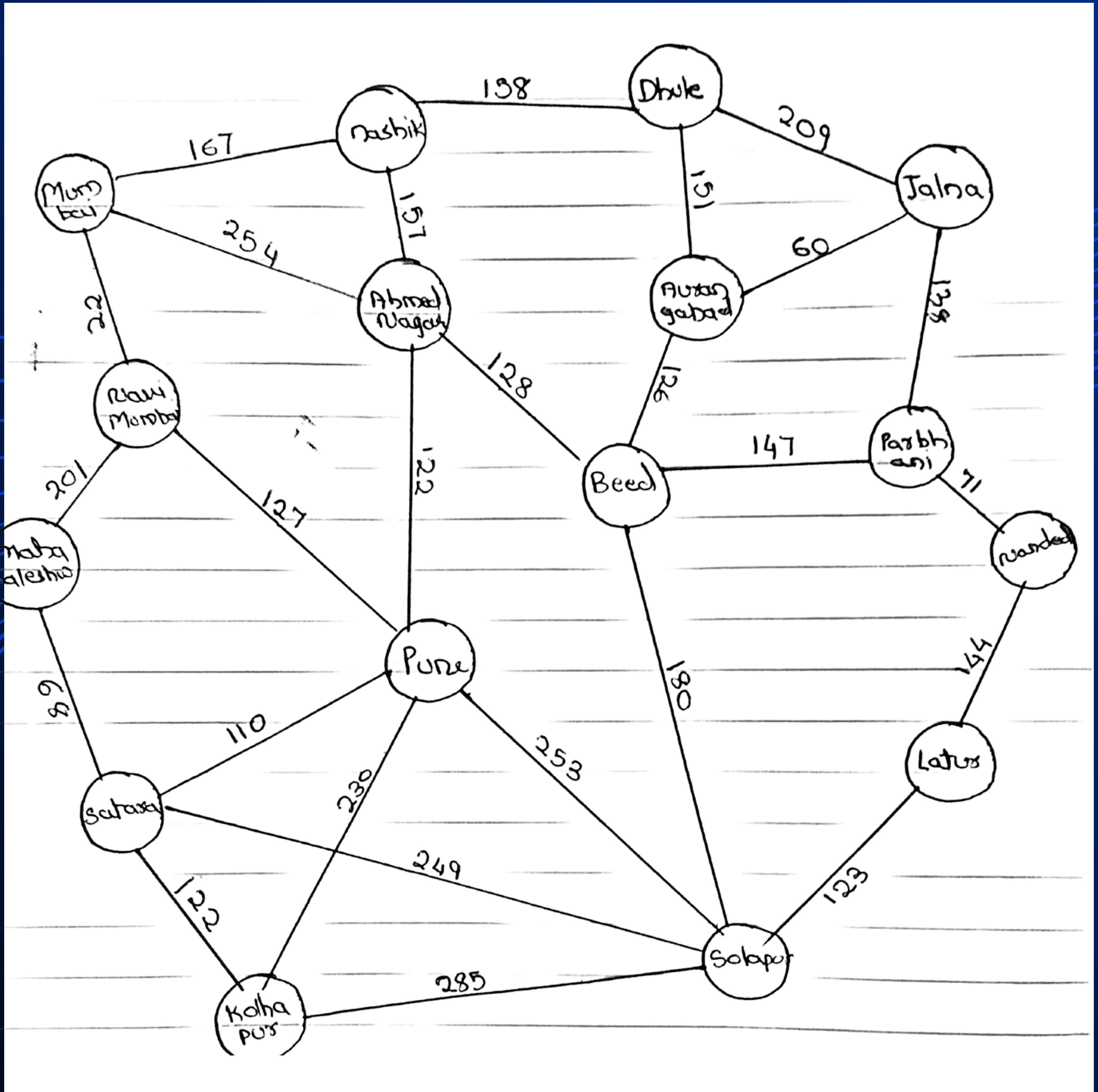
CONS

- It does an obscured exploration that consumes a lot of time while processing,
- It cannot handle negative edges.
- This leads to acyclic graphs and most often cannot obtain the right shortest path.

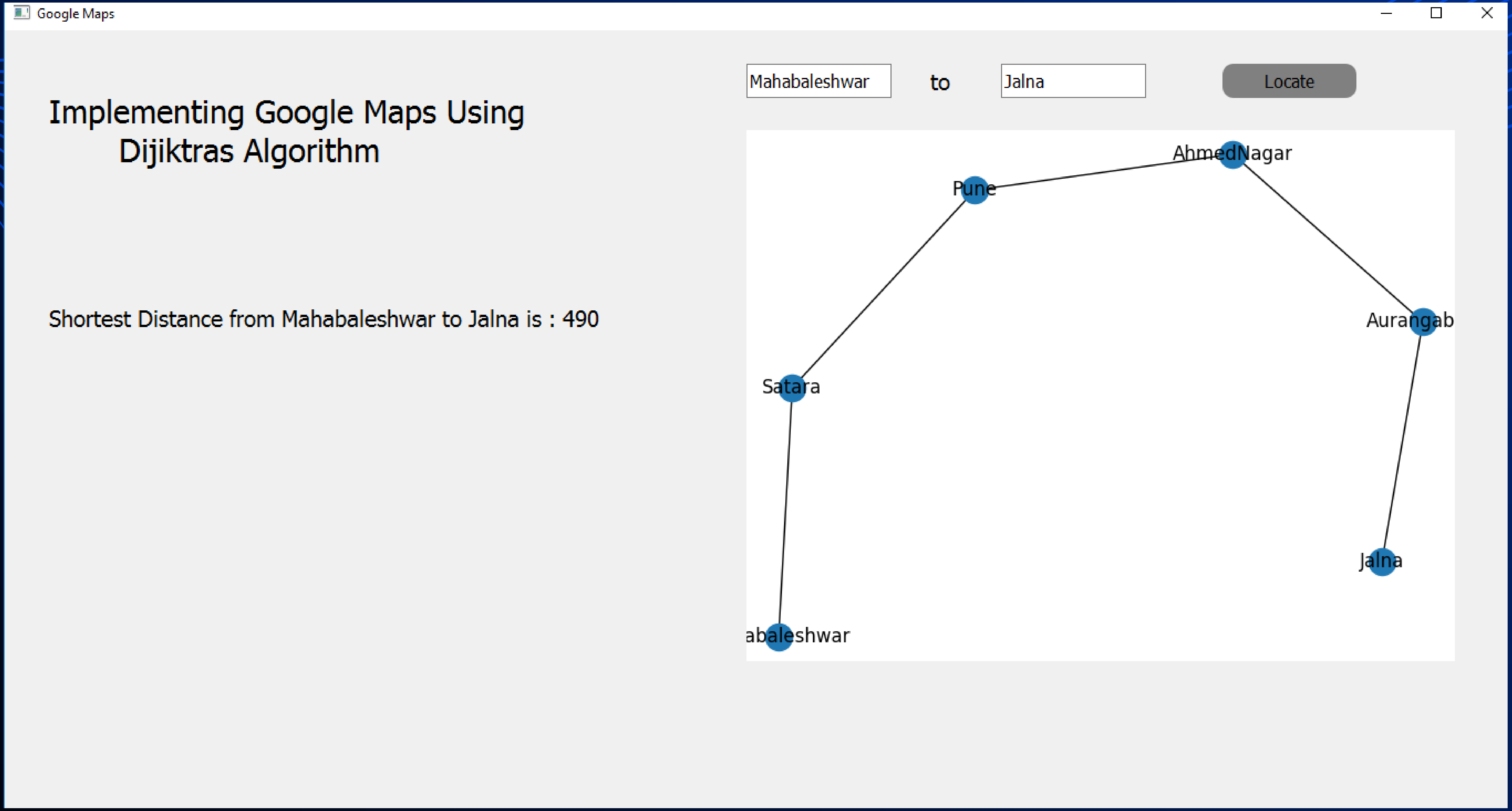


IMPLEMENTATION OF GOOGLE MAPS

SNAPSHOTS


Dhule
138
209
nashik
167
Mum bai
157
151
Jalna
254
60
Ahmed Nagar
Auran gabad
22
134
128
126
Navi Mumbai
147
Parbh ani
Beed
201
127
122
71
Maha baleshwa
nanded
Pune
180
144
69
110
253
Latur
Satara
230
249
123
122
Solapur
285
Kolha pur

SNAPSHOTS



Mahabaleshwar to Jalna

Implementing Google Maps Using
Dijiktras Algorithm

Shortest Distance from Mahabaleshwar to Jalna is : 490

TIME COMPLEXITY

DIJKSTRA'S ALGORITHM COMPLEXITY

- TIME COMPLEXITY: O(E LOG V)
  WHERE, E IS THE NUMBER OF EDGES AND V IS THE NUMBER OF VERTICES.

- SPACE COMPLEXITY: O(V)

FUTURE SCOPE

- We can use another feature of Google Maps i.e Single Source Multiple Destinations but it is little bit complex to implement.
- SSMD is a technique where we can create user defined milestones or stops between Source and Destination.
- This technique can be achieved using Route Optimization algorithm.
- This algorithm just takes single source and more than 2 destinations upto 150 destinations as input and performs some operations and returns optimised path between Source and Destinations.
- This algorithm is widely used by UBER, DOMINOS, and other Delivery System Apps.

RESOURCES USED

SOFTWARE RESOURCES USED

- Programming language and Frameworkls

1. Python Programming Language (3.7)
2. PyQt5 for UI implementation
3. Matplotlib for Graph Visualization
4. Networkx to plot Graph

- VS Code IDE

thank you