

A stylized, abstract illustration of a city map. It features various geometric shapes representing buildings and streets in shades of blue, red, and white. A prominent light blue area in the center represents a body of water, with a small white shape inside it. The overall style is modern and minimalist.

DIJKSTRA'S ALGORITHM

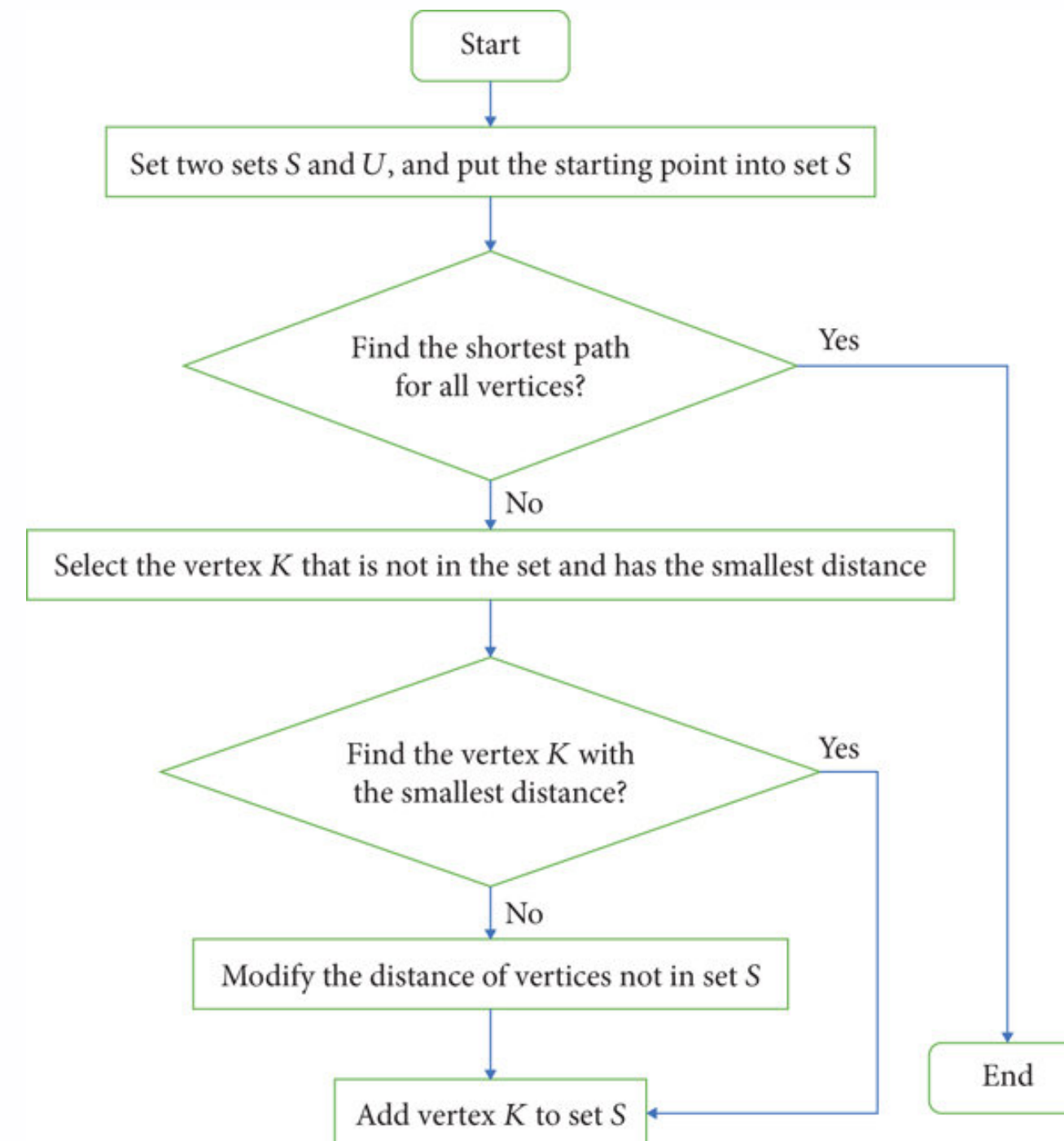
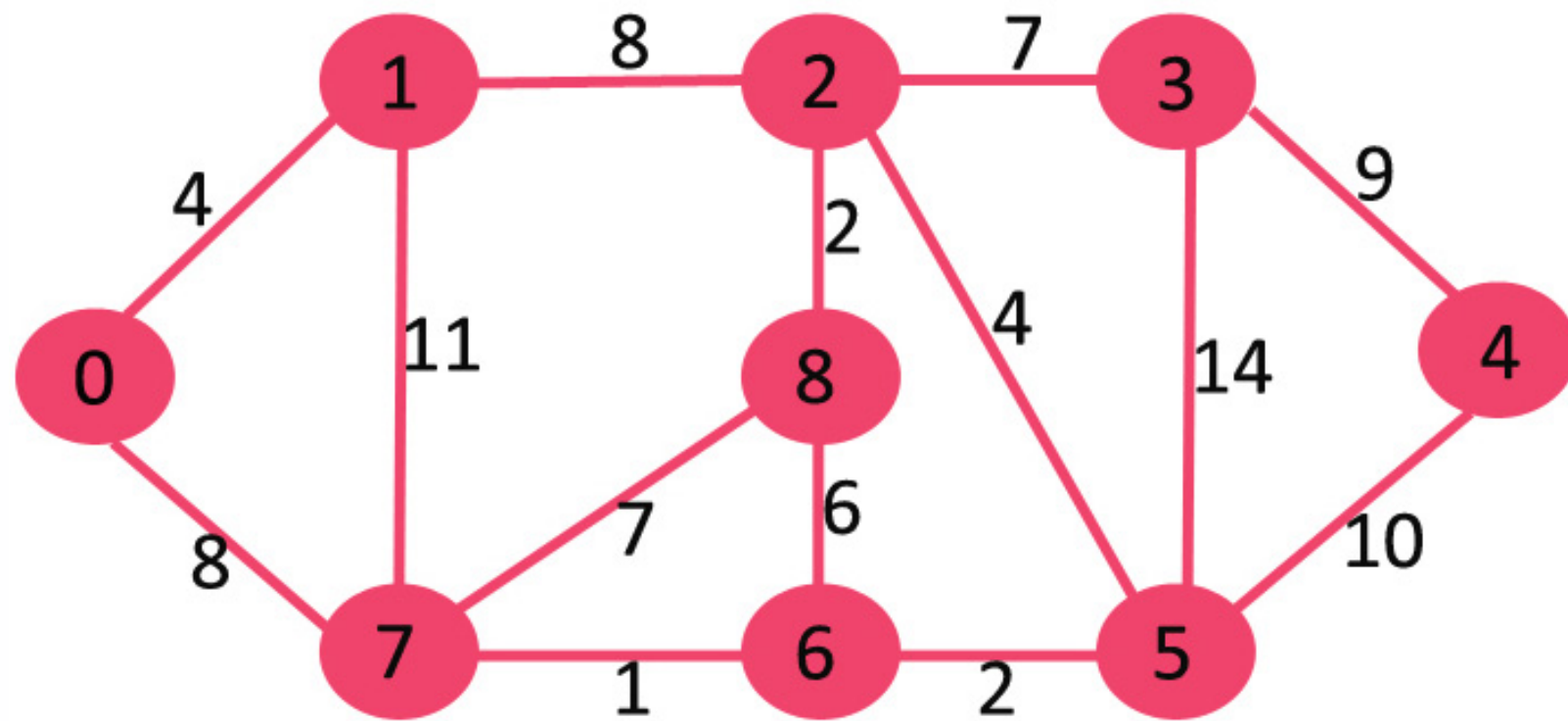
Finding the shortest path
between two points in a
Map using Dijkstra's
Algorithmn

Introduction and Objectives

- Project Overview: The project focuses on finding the shortest path between two geocoordinates using Dijkstra's algorithm.
- Objectives: The main objective is to calculate and visualize the optimal path on a map, considering road networks.
- Technologies Used: Python, OSMnx, NetworkX, Plotly, and NumPy.

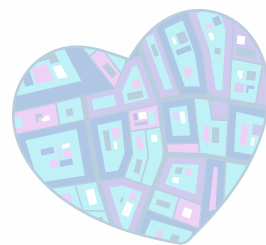
Algorithm/Flowchart

- Dijkstra's Algorithm: This algorithm is employed to find the shortest path between two nodes in a graph



Python Packages and Challenges

- **OSMnx**: Used to retrieve road network graphs from OpenStreetMap.
- **NetworkX**: Utilized for graph representation and manipulation, including Dijkstra's algorithm implementation.
- **Plotly**: Employed for visualizing maps and paths.
- **NumPy**: Used for array operations and mathematical calculations.
- **Challenges/Issues Faced**: The inbuild function `get_nearest_nodes` wasn't working in Osmnx So I had to recreate it. The runtime is too high as it works on real world dataset.



THANK YOU

