19CSE212- DATA STRUCTURES AND ALGORITHMS

Submitted by:

1.BL.EN.U4CSE22268 -Y.TANVI

2.BL.EN.U4CSE22218 -K.DHARA LAKSHMI

3. BL.EN.U4CSE22237-M.DEVA KUMARI

4. BL.EN.U4CSE22249 -R.HARSHA VARDHAN

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

“COMPUTER SCIENCE AND ENGINEERING”

AMRITA SCHOOL OF ENGINEERING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

BENGALURU-560035

INTRODUCTION:

The Map Navigation System based on the Optimal Dijkstra Algorithm is a part of the Intelligent Transportation System (ITS) designed to address the traffic supply and demand contradiction in modern society. This system aims to provide a comprehensive and personalized navigation experience to users, utilizing advanced technologies such as automatic positioning, computer technology, and wireless communication.The current vehicle navigation systems in the market lack efficient information sharing and fail to meet users' personalized and comprehensive requirements. By leveraging the Optimal Dijkstra Algorithm, the Map Navigation System can plan optimal routes for users, considering factors such as traffic conditions and road networks.Scholars have conducted studies to optimize the Dijkstra Algorithm for point path planning, significantly improving the efficiency of the optimal path search process. However, there are still challenges, such as the algorithm's high computational requirements and inability to meet dynamic searching needs in urban road networks.To address these challenges, this proposes an improved Dijkstra Algorithm for the intelligent vehicle navigation system. This system provides features like map management and updating, traffic information query, destination search, shortest path planning, and multimedia playback, offering travelers a convenient and efficient travel service.Overall, the Map Navigation System based on the Optimal Dijkstra Algorithm represents a significant advancement in intelligent transportation systems, providing users with a seamless navigation experience and effectively addressing traffic problems using high and new technologies.

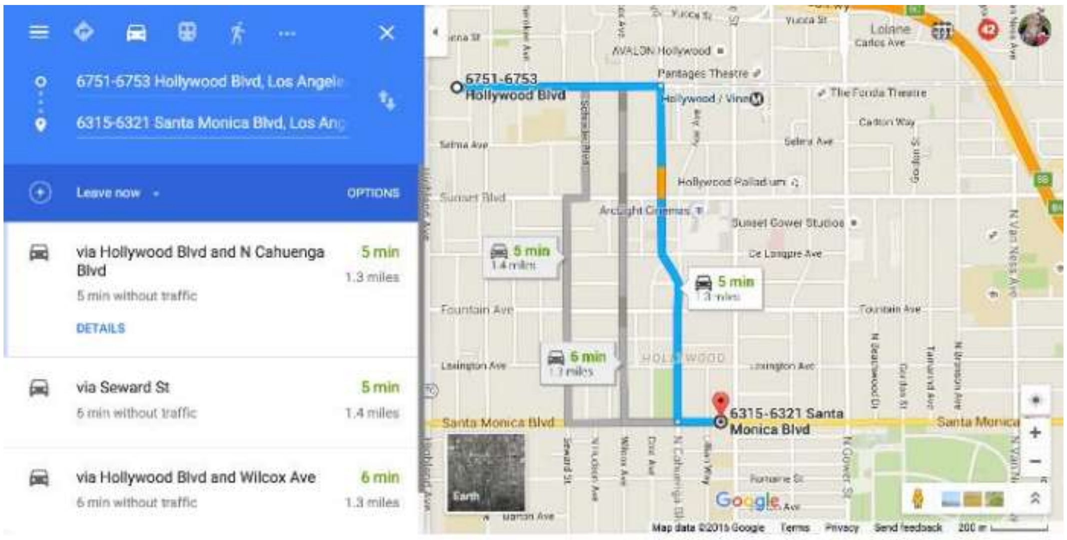
ABSTRACT:

This project presents a comprehensive exploration of graph traversal and path finding algorithms applied to a geographical network. The implementation covers data initialization, connectivity establishment, and efficient exploration methods. The use of Dijkstra's Shortest Path Algorithm, along with specialized approaches for finding the easiest and safest paths, contributes to the optimization of route planning. The project emphasizes the significance of these algorithms in real-world applications, addressing challenges in navigation, risk assessment, and decision-making. The algorithms showcased provide valuable insights for enhancing navigational capabilities within complex networks, facilitating informed decision-making and resource optimization. This work contributes to the advancement of efficient path finding strategies in diverse scenarios. Smart City Routes (SCR) are an important part of the discussion about urban traffic management (UMM). As cities around the world grapple with the challenges posed by rapid urban growth and increasing transportation needs, make a commitment by integrating smart technologies into urban transportation. SCR involves using technology to improve and simplify urban transportation, promoting efficient, cost- effective and customer-friendly solutions. This case study covers various areas of SCR, examining its principles, basic operations, and implications for urban planning and management. By reviewing a wide range of existing studies, this article aims to provide an overview of the current state of SCR measurement, identifying key issues, challenges and opportunities shaping the future of urban mobility.

<u>Main View:</u>

Provides the graphical user interface for route selection and display. The graphical user interface facilitates route selection and display, serving as the user-facing component of the system. However, there is room for improvement in error handling to ensure more graceful responses in scenarios such as missing database files or invalid user input. Enhancements should be implemented to detect and appropriately handle these errors, providing users with clear and informative feedback to guide

them in resolving issues. This improvement in error handling would contribute to a more user-friendly experience, offering better guidance and understanding in situations where unexpected errors or invalid inputs may occur during route selection or database file access.



6751-6753 Hollywood Blvd, Los Angele
6315-6321 Santa Monica Blvd, Los An

Leave now   OPTIONS

via Hollywood Blvd and N Cahuenga Blvd   5 min
5 min without traffic   1.3 miles
DETAILS

via Seward St   5 min
5 min without traffic   1.4 miles

via Hollywood Blvd and Wilcox Ave   6 min
6 min without traffic   1.3 miles

Data Set:

Wayfinding, Path Planning, and Navigation Dataset from Kaggle:

The Map Navigator project leverages datasets sourced from Kaggle, focusing on wayfinding, path planning, and navigation. These datasets offer comprehensive information on outdoor and indoor environments, including road networks, landmarks, and indoor trajectories. The datasets from Kaggle provide valuable insights and data points for developing and testing algorithms related to route optimization, real-time navigation, and location-based services. By utilizing these datasets, the Map Navigator project aims to enhance its functionality and accuracy, offering users a seamless navigation experience both indoors and outdoors.

Data Structure:

For a Map Navigation System based on the Optimal Dijkstra Algorithm, several data structures can be used to efficiently represent and process the geographical network. Here are some key data structures:

1. **Graph:** The geographical network can be represented as a graph, where nodes represent locations (such as intersections or landmarks) and edges represent connections between these locations (roads or paths). This graph can be either directed or undirected, depending on the application.

2. **Priority Queue:** A priority queue is used in Dijkstra's Algorithm to store and retrieve nodes based on their distance from the start node. This allows the algorithm to efficiently select the next node to explore.

3. **Hash Table/Map:** These data structures can be used to store mapping between nodes and their corresponding distances from the start node. This is useful for quickly updating distances during the algorithm's execution

4. **Heap:** A binary heap can be used as a priority queue to implement Dijkstra's Algorithm efficiently. The heap allows for quick insertion and extraction of the node with the minimum distance.

5. **Set:** A set can be used to keep track of visited nodes during the algorithm's execution to avoid revisiting them and to ensure that each node is processed only once.

6. **Array/List:** Arrays or lists can be used to store the actual geographical network data, such as nodes, edges, and their attributes (like distance or weight).

<u>Algorithm:</u>

Dijkstra improved algorithm

Traditional Dijkstra algorithm: It divides the nodes in the graph into two categories, permanent node(P) and temporary node (T). P nodes are those who have found.

Drawbacks:

1. The traditional algorithm uses adjacency matrix to describe the network nodes and the characteristics of the arc, leading to data redundancy and waste of time.
2. The algorithm complexity is too high, it is not suitable for large-scale network shortest path search.
3. In the real-time way, the traffic impedance needs to be considered, in order to improve the accuracy of calculation.
4. The traditional algorithm uses greedy strategy to keep local optimum at each stage, and from the global perspective, the overall result may not be optimal.

Improved algorithm:

1. Use adjacency table to store road network topology.
2. Use binary heap as a priority queue.
3. The traffic impedance analysis.