Comprehensive experiment report

Subway route planning

Content

Participants and assignment

Name	Student ID	Workload
LIU HONGYU	20223802048	100%
QIU GUANSHENG	20223802080	100%
LIU JUNJIE	20223802068	100%

Problem description and basic requirements

Problem description:

We need to design a Python program that allows users to query the subway map of Guangzhou and find the shortest and recommended routes between two stations. This program also needs to have a UI interface and clear and understandable displays.

Basic Requirements:

1. The program should be able to calculate the shortest route between two given stations and provide a recommended solution.
2. All functions should be equipped with a visual interface and UI.
3. The program should display the length of the routes (in terms of number of stops or distance).
4. The recommended routes should be displayed on a map in a browser, with interactive functionality.
5. By simply changing the city code in the map class, the route planning system can handle subway lines in all cities in China.

Extended Functions:

1. The program should be able to calculate the route with the fewest number of stops between two given stations and provide a recommended solution.
2. The program should be able to calculate the route with the fewest transfers between two given stations and provide a recommended solution.

Tools and the knowledge used

1.The program uses an adjacency list graph structure to store the information of each station, representing the graph by storing all adjacent nodes of each node as a linked list.
2.Save the graph data as a JSON file for easy retrieval later, and random weights are used
3. The program uses Dijkstra's algorithm to solve the shortest path calculation problem.
4. The program uses the depth-first search algorithm of the graph to solve the problem of the minimum number of stops.
5. The program uses the dictionary data structure stores the mapping relationship between stations and lines to solve the problem of the minimum transfer times.
6. The program uses PyWebIO to display messages and implement interactive functionality.
7. The program was developed using VsCode and Conda environment.

Analysis and implementation

The task at hand involves planning the shortest route on subway railways between given start and end stations. To efficiently handle this problem, the choice of appropriate data structures and algorithms is crucial. One key consideration is the utilization of an adjacency list to load station information. This approach offers a space complexity of O(n) and proves particularly advantageous for sorting all the outgoing edges of a given station. By employing the adjacency list, the algorithm can efficiently navigate through the subway network, facilitating quick access to relevant station details.

In the context of algorithm selection, the use of a priority queue becomes essential. When updating the shortest circuit of a particular station multiple times, it's imperative to note that elements inserted during previous updates cannot be deleted or modified. This constraint results in a priority queue with a fixed number of elements (0(m)), leading to a time complexity of O(mlogm). The efficiency of Dijkstra's algorithm, implemented with a priority queue, shines in sparse graph scenarios (where m = O()). In such cases, it outperforms the Bellman-Ford algorithm. Conversely, in dense graphs (where m = O(n^2)), a brute-force approach becomes more favorable than using a priority queue.

The implementation of the chosen algorithm involves several key functions. Notably, the Dijkstra algorithm's weight can be modified to influence the planning element, allowing for a trade-off between the shortest distance and the least number of stations. The distance between two stations is calculated using Mercator coordinates from Baidu Map, ensuring accurate and real-world distance metrics. The user interface is designed using the pywebio package, enabling the creation of an interactive interface on the browser through Python code. For visualizing the subway route, the plotly package is employed to display various elements, such as subway lines, stations, and the city map.

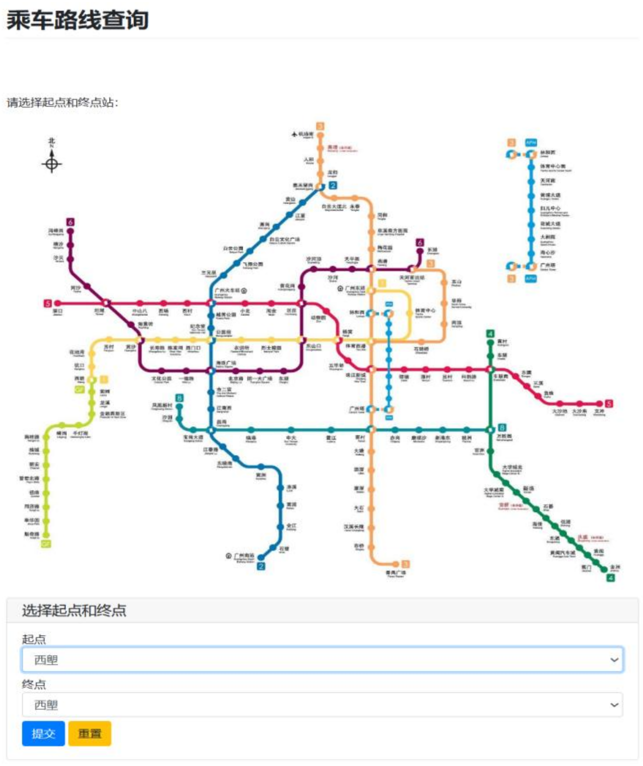
Furthermore, the system incorporates functionality for planning routes with the least transit stations. This is achieved by implementing a Breadth-First Search (BFS) algorithm on the graph using a queue. This ensures an efficient exploration of the subway network to identify routes that minimize the number of station transfers, enhancing the user experience.

The effective combination of data structures, algorithms, and implementation details is crucial in developing a robust system for planning subway routes. The careful selection of adjacency lists, priority queues, and algorithmic approaches tailored to the characteristics of the subway network ensures optimal performance and responsiveness.

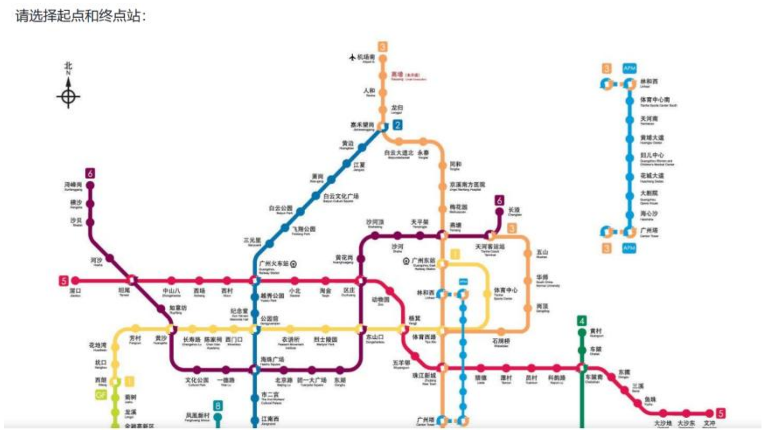
The incorporation of user-friendly interfaces and visualization tools enhances the overall usability and provides a comprehensive solution for navigating subway systems with varying preferences, be it the shortest distance or the least transit stations.

Testing and conclusion

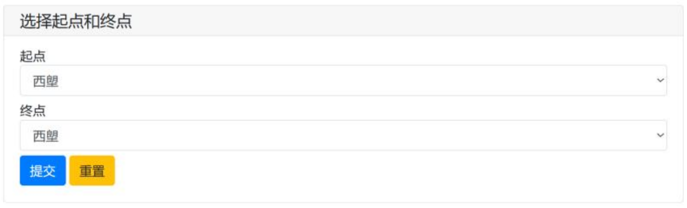
1.Run the control.py file located in the graph directory to start the program. Your browser will open the following page automatically:



2. Hold down Ctrl + mouse wheel to zoom in and out of the map



3. This is User input screen, you can select your start and end stations in the drop-down menu



选择起点和终点

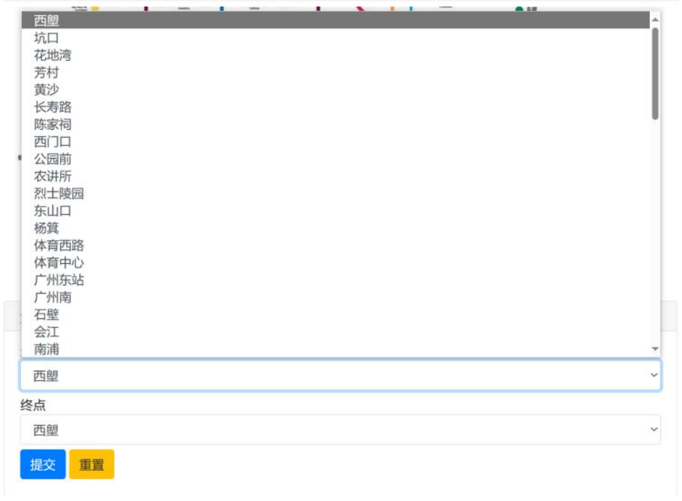
起点

西塱

终点

西塱

提交 重置



西塱
坑口
花地湾
芳村
黄沙
长寿路
陈家祠
西门口
公园前
农讲所
烈士陵园
东山口
杨箕
体育西路
体育中心
广州东站
广州南
石壁
会江
南浦

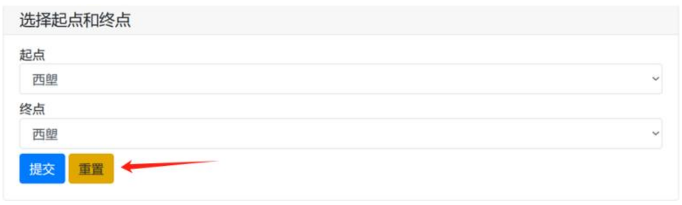
西塱

终点

西塱

提交 重置

4. When you have entered the wrong station name, you can click the reset button to quickly clear the station name you have entered, or you can simply re-select it.



选择起点和终点

起点

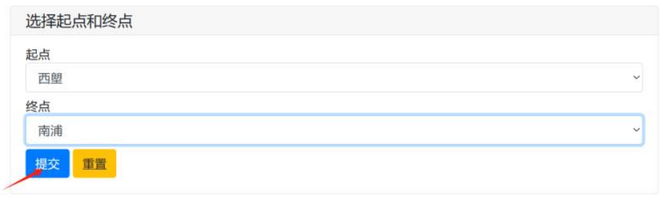
西塱

终点

西塱

提交 重置

5. Click on the "Submit" button to select your plan.



选择起点和终点

起点

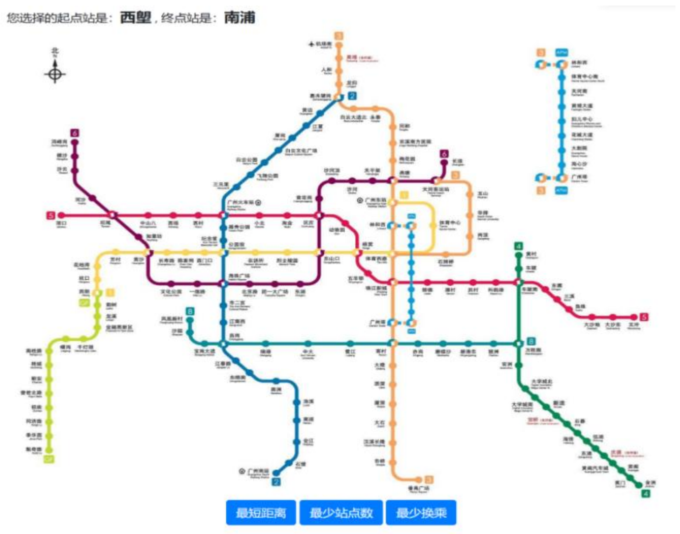
西塱

终点

南浦

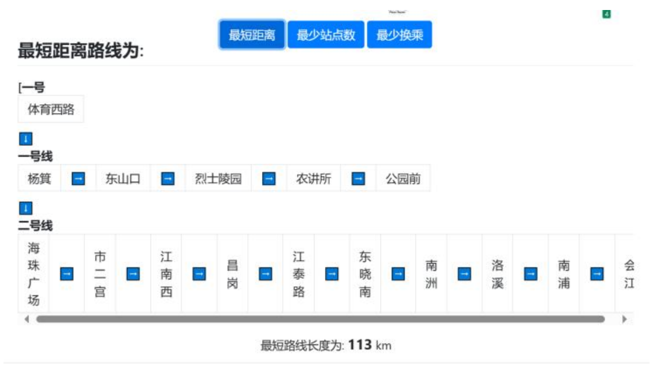
提交 重置

6. After submitting, you can choose the metro route option you want (shortest distance, minimum number of stops, minimum transfers)



您选择的起点站是：西塱，终点站是：南浦

7. If you choose the shortest distance option, the programme selects routes based on the distances travelled by different routes.



最短距离路线为:

[一号

体育西路

↓

一号线

杨箕 → 东山口 → 烈士陵园 → 农讲所 → 公园前

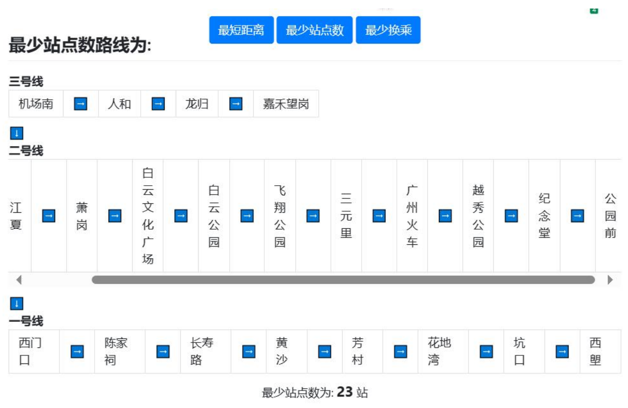
↓

二号线

海珠广场 → 市二宫 → 江南西 → 昌岗 → 江泰路 → 东晓南 → 南洲 → 洛溪 → 南浦 → 会江

最短路线长度为: 113 km

8. If you choose the Minimum Number of Stops option, the programme will select the route based on the number of stops passed. (Take Airport South – Xi Lang as an example)



最少站点数路线为:

三号线

机场南 → 人和 → 龙归 → 嘉禾望岗

↓

二号线

江夏 → 萧岗 → 白云文化广场 → 白云公园 → 飞翔公园 → 三元里 → 广州火车 → 越秀公园 → 纪念堂 → 公园前

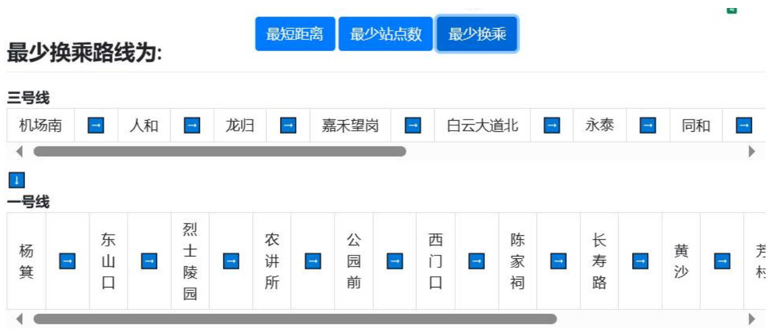
↓

一号线

西门口 → 陈家祠 → 长寿路 → 黄沙 → 芳村 → 花地湾 → 坑口 → 西塱

最少站点数为: 23 站

9. If you choose the Least Interchange Option, the programme will select the number of interchanges according to the number of interchanges on different routes (Take Airport South – Xi Lang as an example)



最少换乘路线为:

三号线

机场南 → 人和 → 龙归 → 嘉禾望岗 → 白云大道北 → 永泰 → 同和 →

↓

一号线

杨箕 → 东山口 → 烈士陵园 → 农讲所 → 公园前 → 西门口 → 陈家祠 → 长寿路 → 黄沙 →

10. If you want to choose another line, you can restart the programme

Conclusion: This programme completes the tasks required by the teacher well, exercises the group's coding ability and deepens the group's understanding of the subject of data structures and algorithms.

Summary

While algorithms and data structures show relative efficiency, employing Double Breadth First Search can diminish the algorithm's time complexity when planning routes with minimal transfer stations. In the context of algorithm selection, it is important to recognize the advantages of specific algorithms such as Dijkstra's algorithm for sparse graphs, with lower time complexity compared to the Floyd-Warshall algorithm. This choice of algorithms contributes to the overall efficiency of the system.I find the system particularly impressive due to its more intuitive interactive mode, allowing for the real-time drawing of circuit diagrams. Additionally, the system easily transitions to the subway line map for all cities in the country, showcasing its high reusability.

Moreover, the system includes an encapsulated line drawing device that facilitates swift reuse for drawing various types of diagrams beyond subway maps. the course's significance is underscored by the valuable insights gained during the project. For instance, the exploration of mapping principles, including the application of hash tables, provides a deeper understanding of the underlying mechanisms. Teaching this course in C/C++ is advocated due to the perceived drawbacks of Python's excessive encapsulation, which can impede the swift implementation of algorithms. In essence, the course not only imparts technical knowledge but also equips students with practical skills applicable to real-world scenarios.

GitHub repository address: https://github.com/david188888/SubwayRoute-management