12/14/2018

Chen Wang

Fudan University

Data Structure and Algorithm

Project Documentation

Table of Contents

[I. Algorithm Adopted in the application 2](#_Toc532541562)

[II. Performance analysis based on the running time 2](#_Toc532541563)

[1. Single operation time cost 2](#_Toc532541564)

[2. Time cost with varied data volume 4](#_Toc532541565)

# Algorithm Adopted in the application

The aim of the project is to implement a subway navigator based on the subway map in Shanghai. Both the subway map and the bus map are represented by a group of vertices and edges, so that the algorithms based on the graph theory can be applied.

All the route planning methods are adapted on the basis of the Dijkstra algorithm. Considering the weight of the edges in the graph are the running time of the subway, the weight of all the edges are positive. Meanwhile, we should notice that there are loops in the graph, hence making the algorithms Breadth-first search and Dag shortest paths inapplicable. Under such constraints, the best algorithm is the Dijkstra algorithm with a complexity of .

The first requirement is to find the route with the least walking time. The implementation of this algorithm is simple because only the stations that are closest to the starting point and the end point are selected as the source and the end point of the graph. Then these two vertices are sent to the Dijkstra algorithm for searching a shortest path with weight.

The second requirement is to find the least transfer route. In order to accomplish this task, a range of stations are searched around the start point and the end point. Here considering the constraint of the walking distance in reality, the stations that are 4 kilometers around the input address are taken into consideration. In the cases that there is no station within the distance of 4 kilometer, then the station with the least walking distance is selected. As to the bus stations in this case, I have noted that in the center of Shanghai, there may be up to hundreds of bus stations within the range of 4 kilometer of a selected address. In this case, only the five nearest stations are taken into consideration so as to save the algorithm running time.

The third requirement is to find the route with the least total time. In this case, I have adopted a similar strategy as the previous case. The total time of subway/bus and walking is recorded for a list of routes. Then the route with the least total time is selected as the final plan.

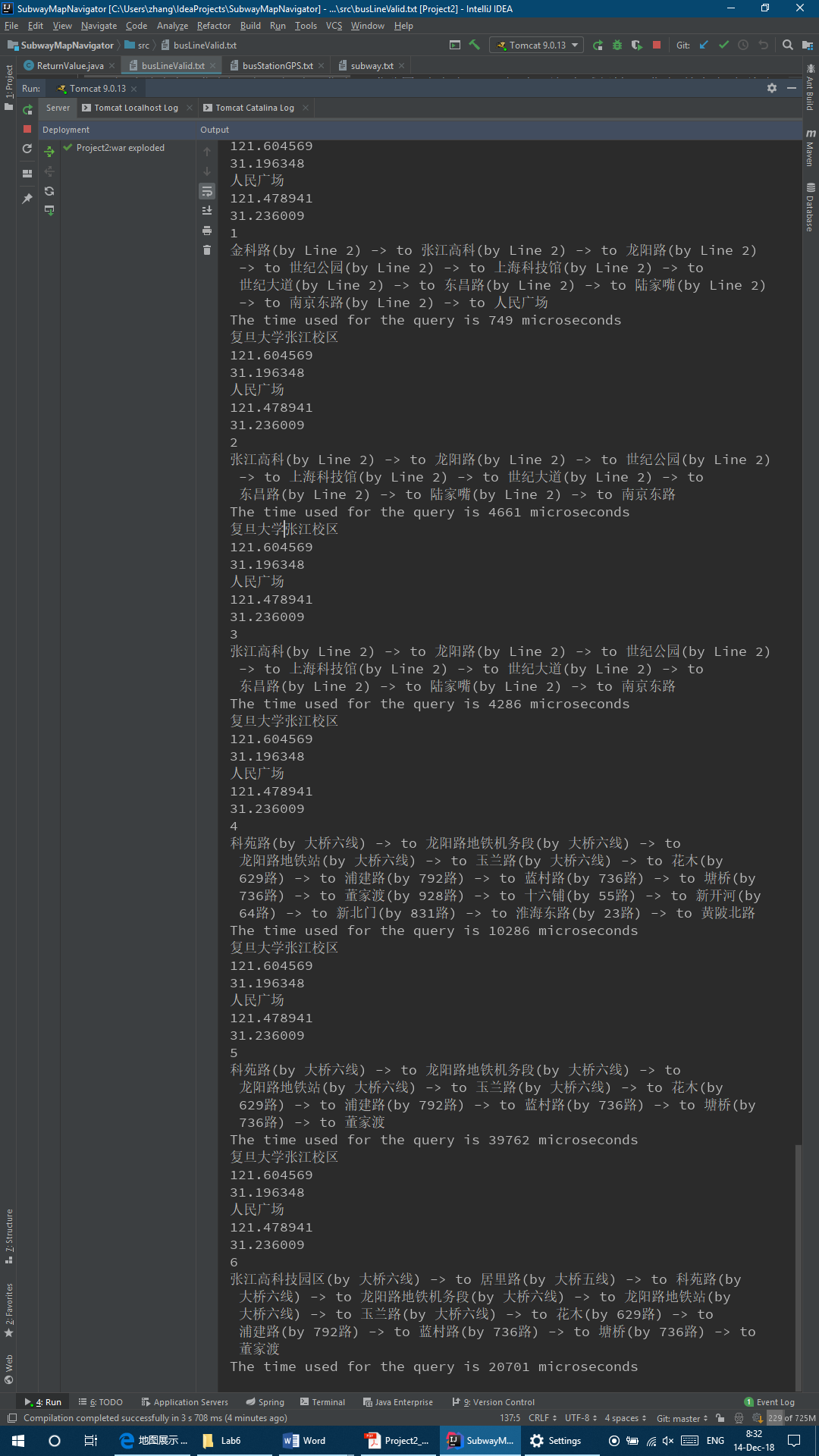
# Performance analysis based on the running time

## Single operation time cost

For a consecutive sequence of single request operation, we can see that the operation time is as the table shown below.

|  |  |
| --- | --- |
| Single Query | Time consumed (milliseconds) |
| 地铁步行最少 | 0.7 |
| 地铁换乘最少 | 4.7 |
| 地铁时间最短 | 4.3 |
| 公交步行最少 | 10.1 |
| 公交换乘最少 | 39.7 |
| 公交时间最短 | 20.7 |

Before analyzing the running time of the algorithm, I would like to clarify the data related to the two graphs. The vertices of the subway in Shanghai is 325 in total, the vertices of bus stations are 2172 in total. Because the graph characteristics of the subway map and the bus map is similar- they are all organized in lines and the density of vertices per line are similar. Therefore, the number of vertices might well represent the complexity of the graph. Considering the time complexity of the Dijkstra algorithm, the time consuming of the bus line query should be about ten times longer than the subway line query. Observing the data above, the experimental result adheres to the theoretical result.



## Time cost with varied data volume

With the automated testing file generating method, we can easily measure the running time of the program on different scale of input data volume. Here some examples are input and the time is analyzed to show that the algorithms has a complexity of .

For the first step, we control the volume of total vertices at , and we can get the following graph by changing the data size of edges. The curve is the tested data got from the running time of the algorithm with each input graph size. After getting all the data and drawing the curve line, I have also made a fitting straight line to show that the relationship between the variable and the value is a linear relationship.

For the second step, I am going to verify that the running time of the algorithm has a logarithm relationship with the variable V of the graph.

From the two pictures above, we can draw the conclusion that the running time of the algorithm used in this project has a time complexity of .