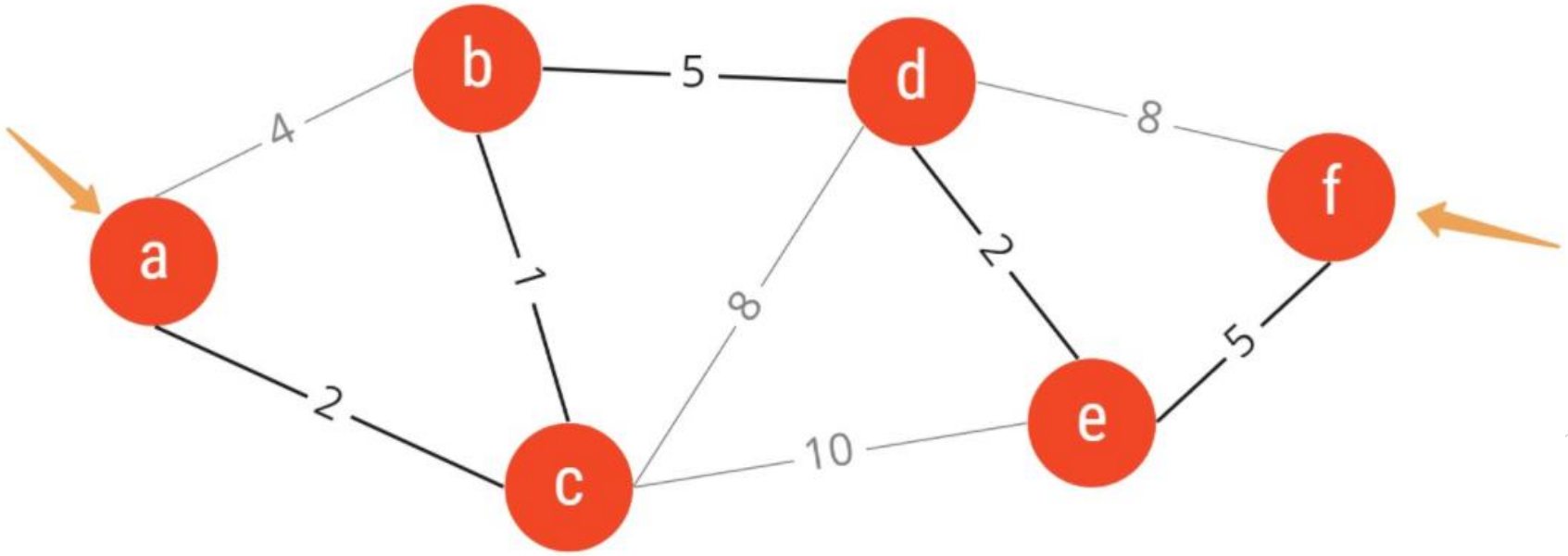
HACETTEPE UNIVERSITY
GEOMATICS ENGINEERING
2020-2021 SPRING SEMESTER
GMT 352 - GEOGRAPHICAL INFORMATION SYSTEMS

ACM SIGSPATIAL GIS Cup – SHORTEST PATH

What is the shortest path problem ?

- The definition of the shortest path problem is to find a path between two nodes in a data set that minimizes the sum of the weights of the edges forming it.



The most important algorithms for solving this problem are;

Dijkstra’s algorithm solves the single-source shortest path problem with non-negative edge weight.

Bellman–Ford algorithm solves the single-source problem if edge weights may be negative.

A* search algorithm solves for single-pair shortest path using heuristics to try to speed up the search.

Floyd–Warshall algorithm solves all pairs shortest paths.

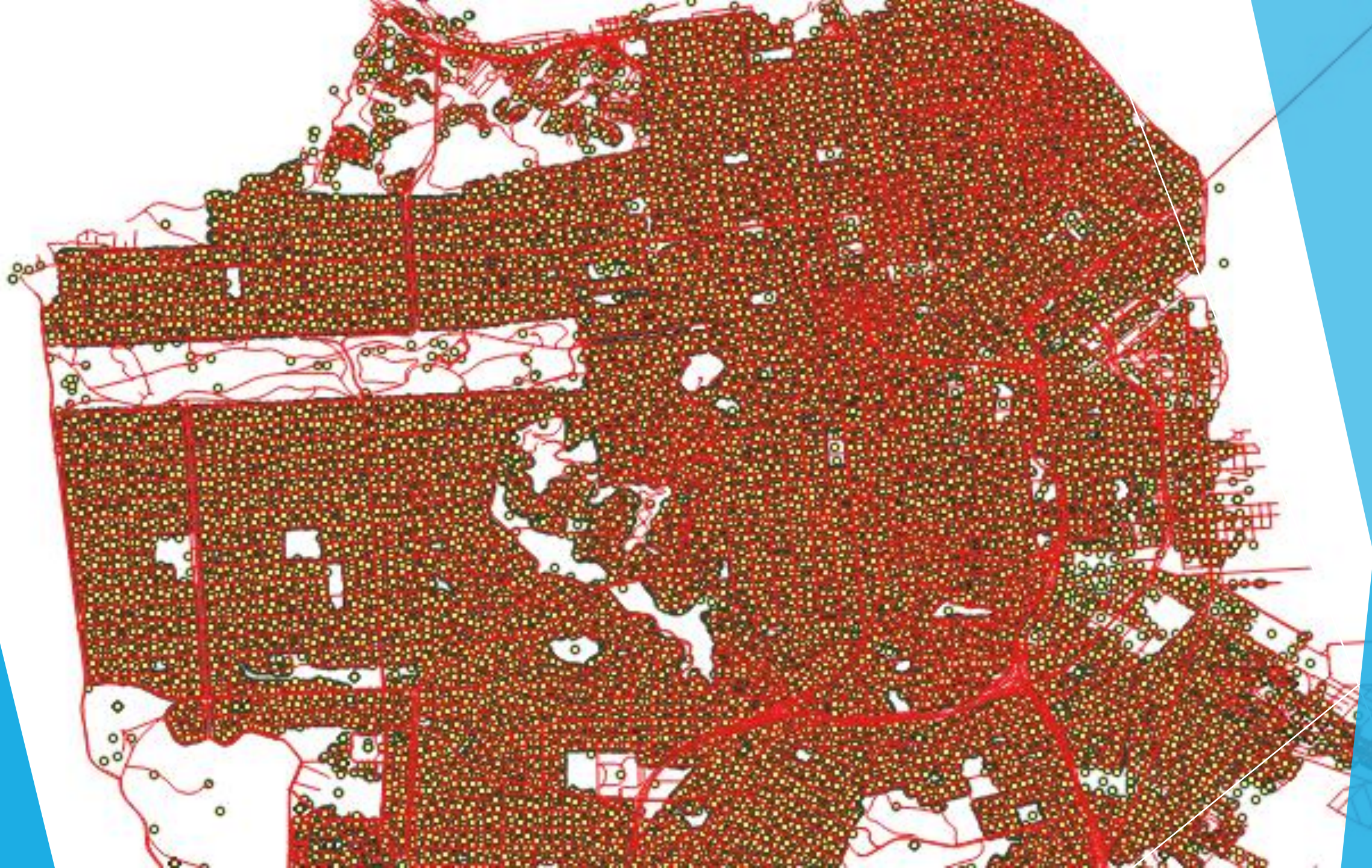
Johnson's algorithm solves all pairs shortest paths and may be faster than Floyd–Warshall on sparse graphs.

Viterbi algorithm solves the shortest stochastic path problem with an additional probabilistic weight on each node.

- Dijkstra's algorithm is an algorithm used to find the shortest path between nodes. Google Maps is used in OSPF (Open Shortest Path First) protocol, game programming, transportation networks. The main purpose of the algorithm is to calculate how to get from one node to another at the cheapest cost. We decided to build our project based on the Dijkstra's algorithm because it is more widely used among these algorithms and more suitable for our project when we researched it.

Finding and processing datasets

- When using San Francisco data, we used other road and node data from this city, as we need the attributes of the roads to be able to examine whether the roads are single lane or double lane.
- We traded between two points using road data and address data.
- We checked the accuracy of the road data on open street map and when we saw that it was correct, we decided to use this road data. Each side of the roads has an id, and the properties of the roads are included. In our code, the case of the road being one or two-way is also used.
- In the address file, we used the point cloud of San Francisco. This data file contains the latitude longitude value and id of each point.




Presidio
Presidio Hill
US 101
Lafayette Heights
Turk Street
San Francisco
Fell Street
Ignatius Heights
Alamo Heights
Corona Heights

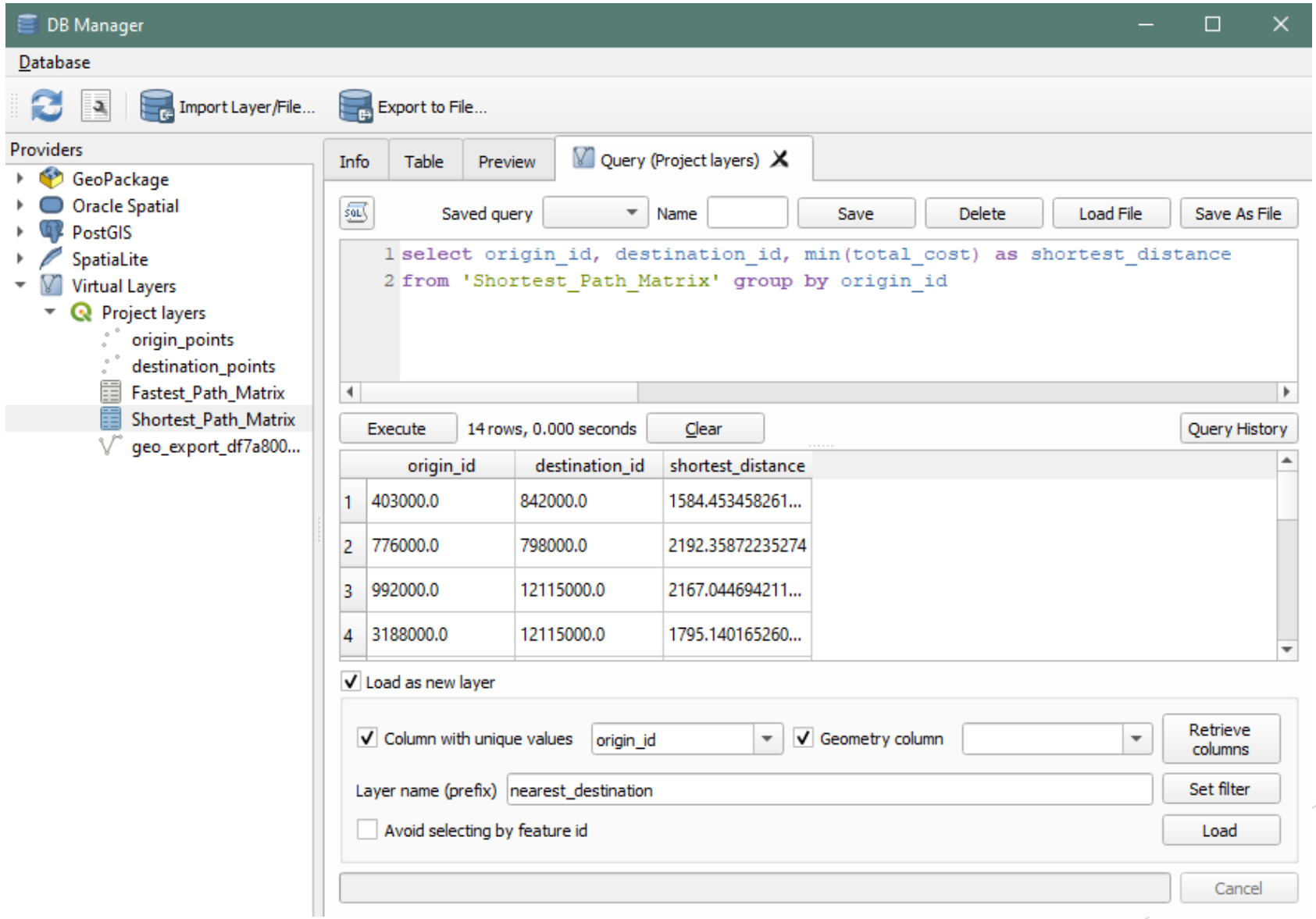
First of all, we determined random 15 starting and 15 ending points from the address data in QGIS.

Shortest_Path_Matrix — Features Total: 225, Filtered: 225, Selected: 0

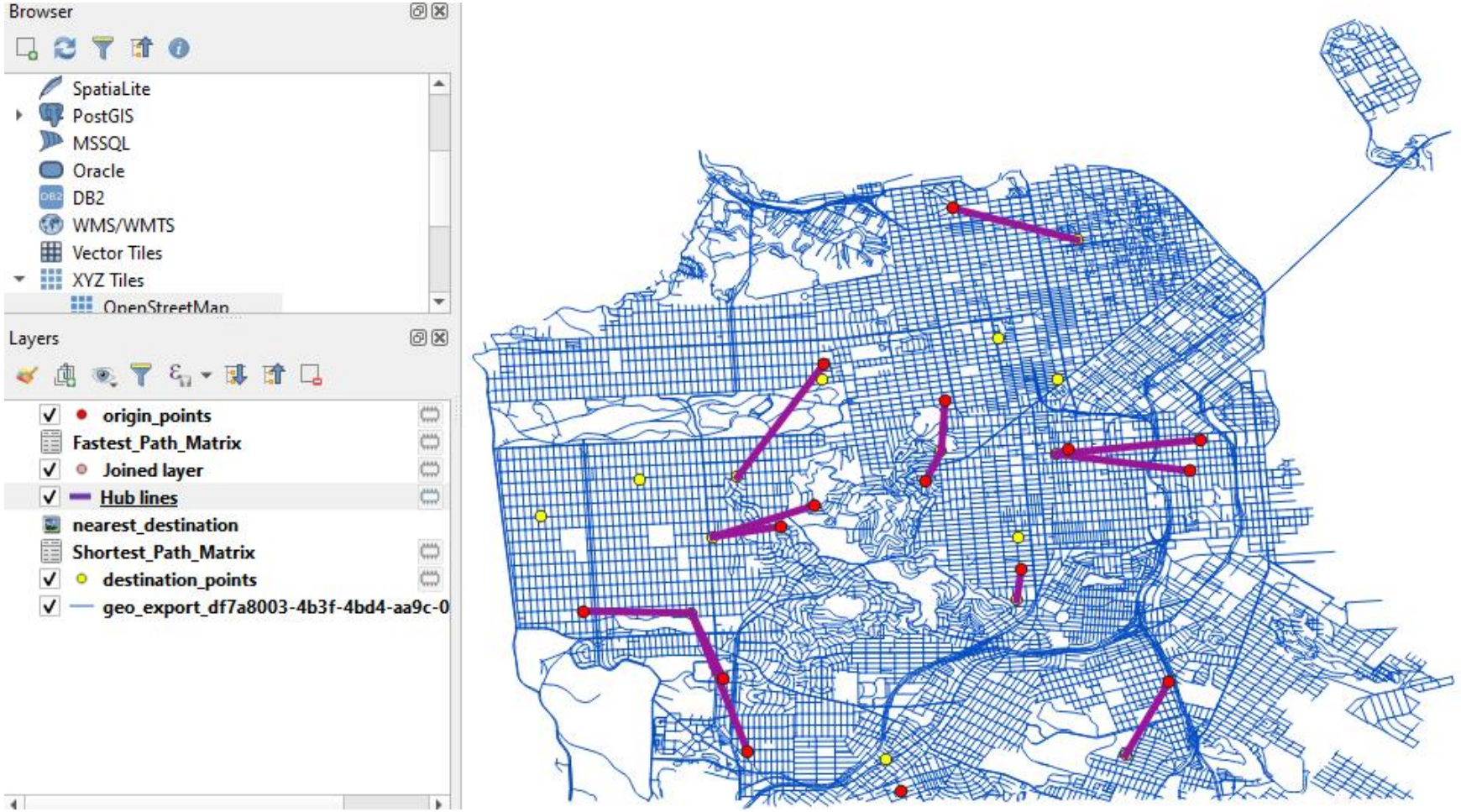
	origin_id	destination_id	entry_cost	network_cost	exit_cost	total_cost
1	403000	798000	25.0832649	7353.3960249	17.3458307	7395.8251205
2	776000	798000	25.8894469	2149.1234448	17.3458307	2192.3587224
3	992000	798000	17.8300207	3560.3756855	17.3458307	3595.5515368
4	3188000	798000	23.1474635	3296.8605529	17.3458307	3337.3538471
5	3258000	798000	NULL	NULL	NULL	NULL
6	4035000	798000	27.8416758	7110.1248240	17.3458307	7155.3123306
7	8041000	798000	24.4465415	8470.0973243	17.3458307	8511.8896965
8	8636000	798000	22.1370662	7527.1999187	17.3458307	7566.6828156
9	9116000	798000	43.9002916	216.2846922	17.3458307	277.5308146
10	11378000	798000	23.5907398	7551.8742900	17.3458307	7592.8108605
11	11654000	798000	12.8154598	10077.7303687	17.3458307	10107.8916592
12	12229000	798000	23.7889377	9091.6347864	17.3458307	9132.7695548
13	12752000	798000	26.7432267	8488.8678627	17.3458307	8532.9569201
14	13265000	798000	27.7139223	10616.9604353	17.3458307	10662.0201883
15	13694000	798000	33.1440462	3315.9144644	17.3458307	3366.4043413
16	403000	842000	25.0832649	1530.3972142	28.9729792	1584.4534583

- After creating the origin and destination points, we created the origin-destination matrices (OD Matrices) of these points. The total cost part in this picture gives the distance between two points.

- We wrote a command that separates the shortest distances between the points in the database manager.



Create the Hub lines



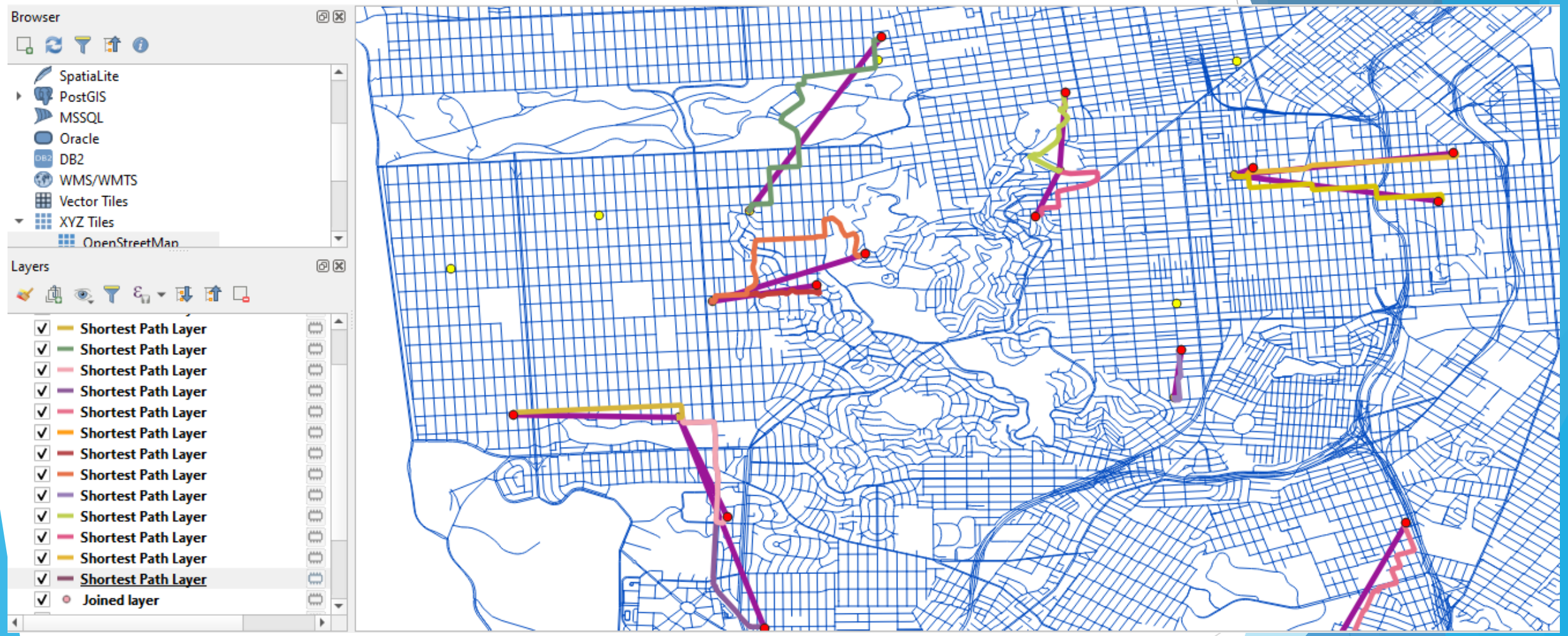
- We drew the hub line according to the distances we found in the shortest path matrix between the start and end points.

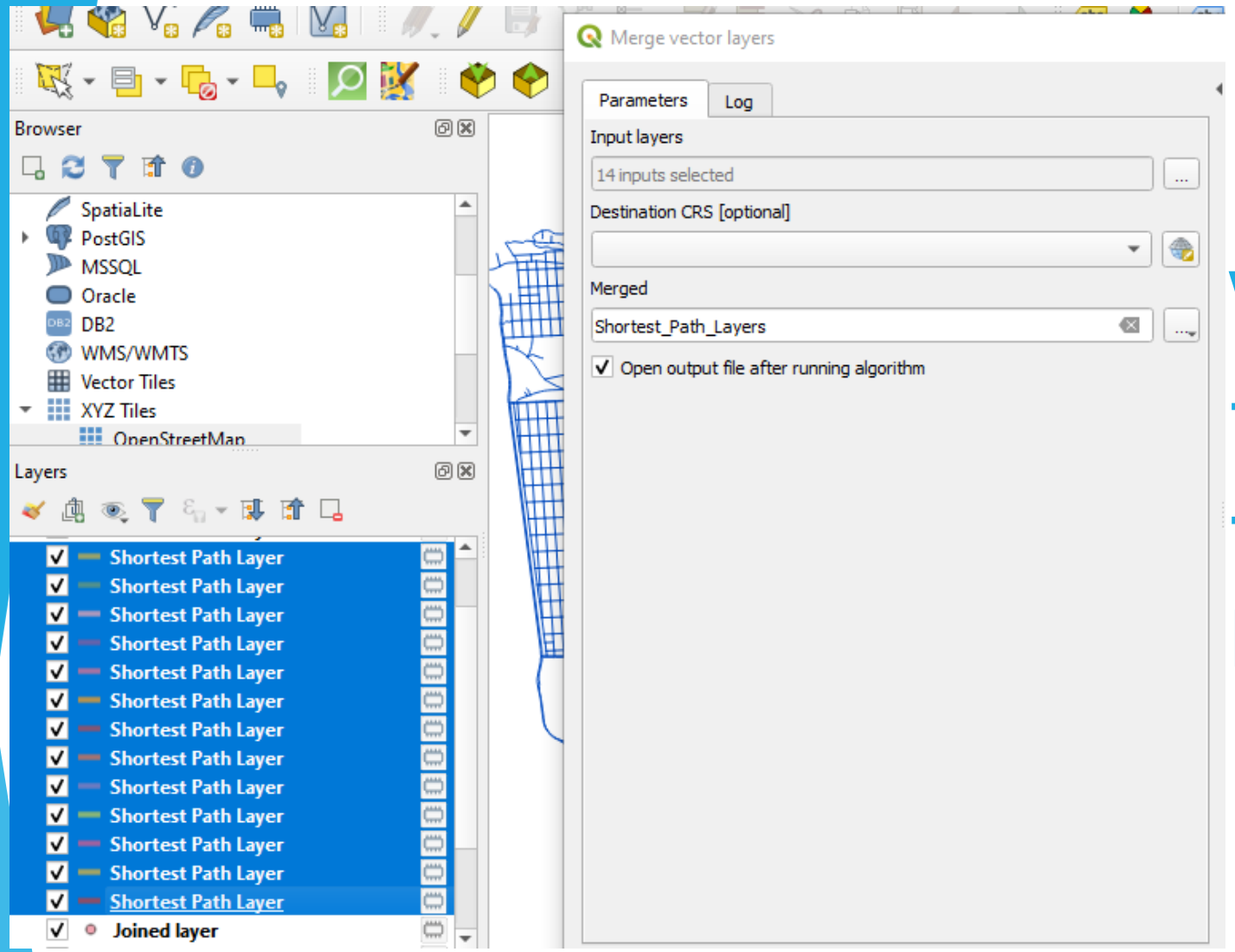
Using this code, we have plotted the shortest distances between points.

```
destination_layer = QgsProject.instance().mapLayersByName('destination_points')[0]
matrix = QgsProject.instance().mapLayersByName('nearest_destination')[0]

for f in matrix.getFeatures():
    origin_expr = QgsExpression('cnn={}'.format(f['origin_id']))
    destination_expr = QgsExpression('cnn={}'.format(f['destination_id']))
    origin_feature = origin_layer.getFeatures(QgsFeatureRequest(origin_expr))
    origin_coords = [(f.geometry().asPoint().x(), f.geometry().asPoint().y())
        for f in origin_feature]
    destination_feature = destination_layer.getFeatures(QgsFeatureRequest(destination_expr))
    destination_coords = [(f.geometry().asPoint().x(), f.geometry().asPoint().y())
        for f in destination_feature]
    params = {
        'INPUT':'geo_export_df7a8003-4b3f-4bd4-aa9c-0bae12b615f9',
        'START_POINT':'{},{}'.format(origin_coords[0][0], origin_coords[0][1]),
        'END_POINT':'{},{}'.format(destination_coords[0][0], destination_coords[0][1]),
        'STRATEGY':0,
        'ENTRY_COST_CALCULATION_METHOD':0,
        'DIRECTION_FIELD':'oneway',
        'VALUE_FORWARD':'B\n',
        'VALUE_BACKWARD':'T\n',
        'VALUE_BOTH':'',
        'DEFAULT_DIRECTION':2,
        'SPEED_FIELD':None,
        'DEFAULT_SPEED':5,
        'TOLERANCE':0,
        'OUTPUT':'memory:'}
    print('Executing analysis')
    processing.runAndLoadResults("qneat3:shortestpathpointtopoint", params)
```

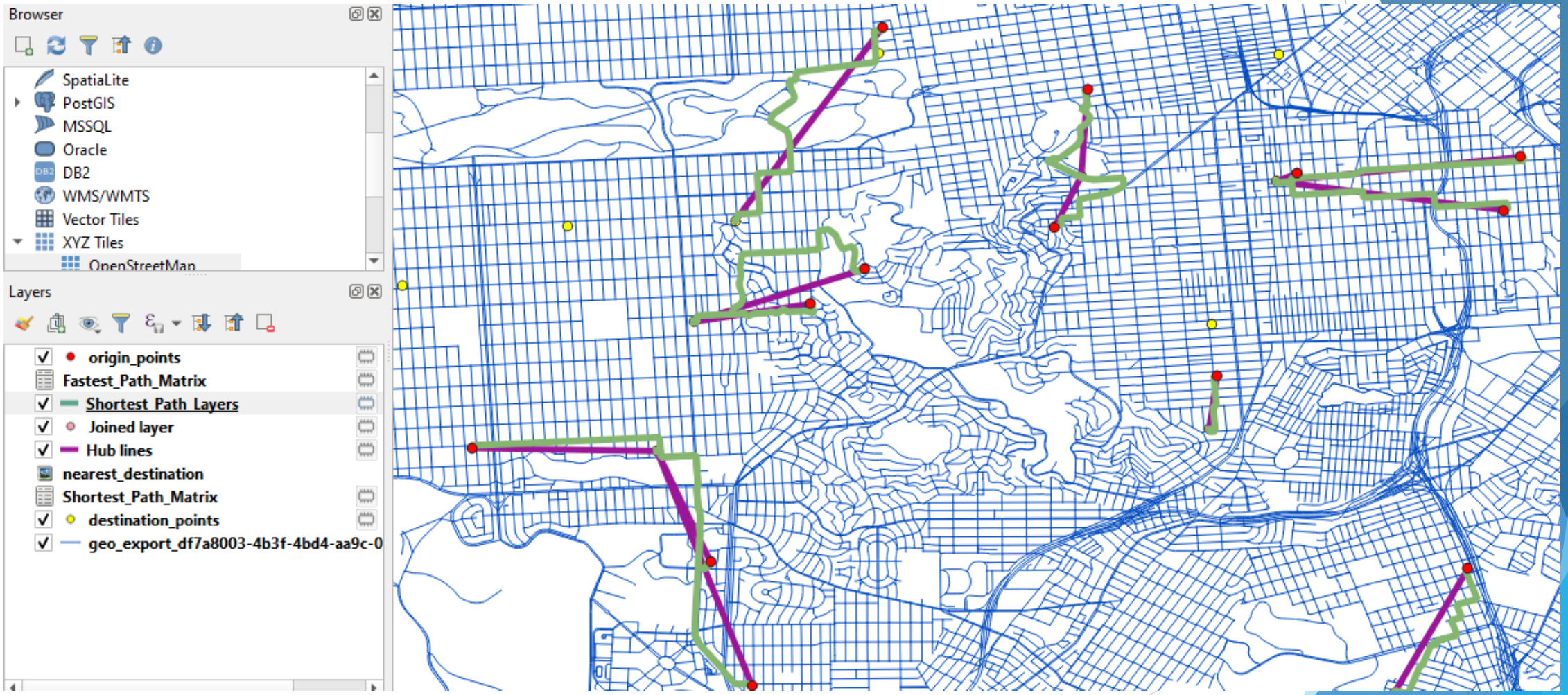
The outputs we get after the code



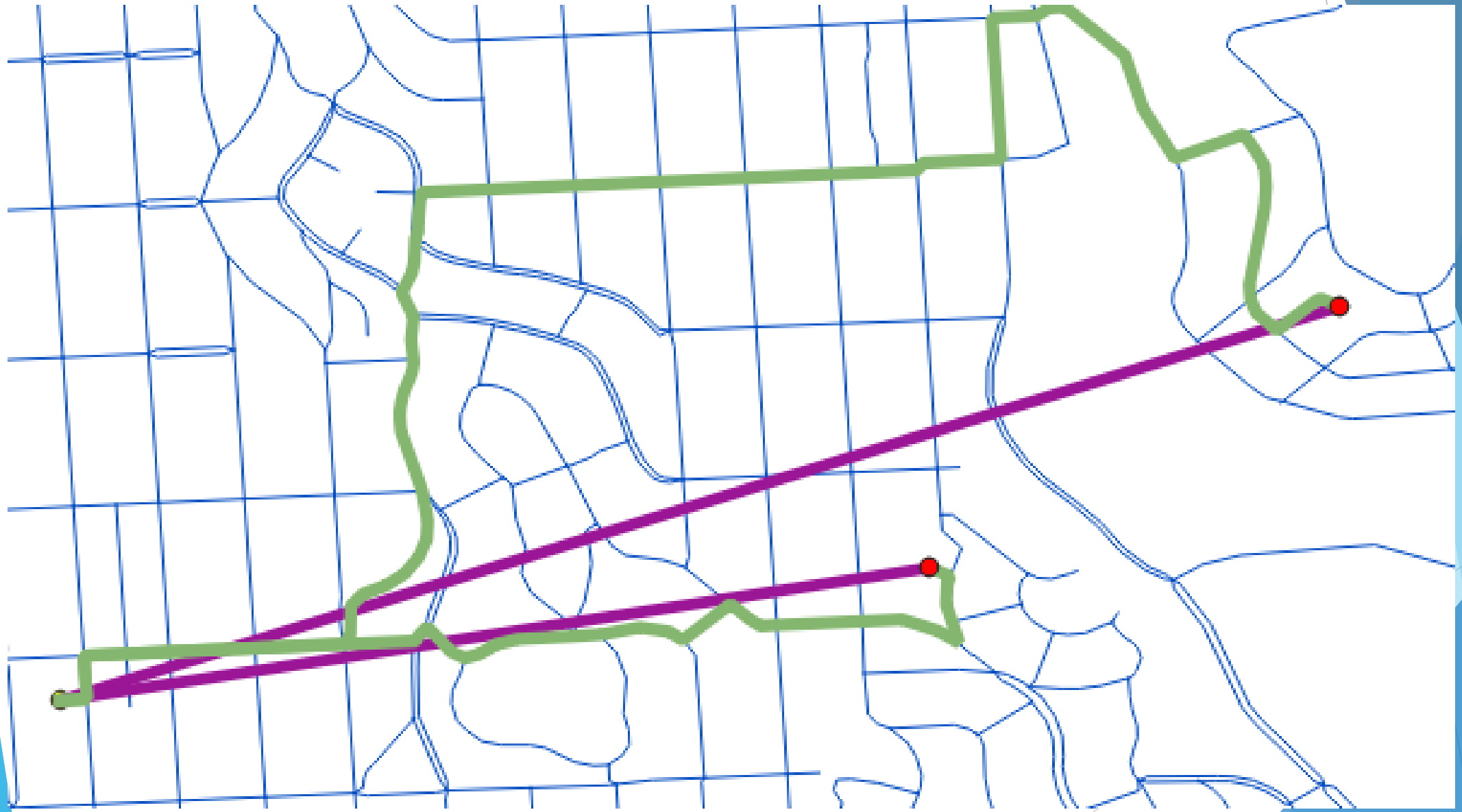


We combined these paths into a single layer.

Merged layer for shortest path



Our Outputs;



REFERENCES;

- https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm#:~:text=Dijkstra's%20algorithm%20(%2F%CB%88da%C9%AA,algorithm%20exists%20in%20many%20variants. (08.06.2021)
- https://en.wikipedia.org/wiki/Shortest_path_problem (07.06.2021)
- https://docs.qgis.org/3.4/en/docs/user_manual/processing/toolbox.html (01.06.2021)
- https://data.sfgov.org/Geographic-Locations-and-Boundaries/Streets-Active-and-Retired/3psu-pn9h (28.05.2021)
- https://data.sfgov.org/Geographic-Locations-and-Boundaries/Addresses-Enterprise-Addressing-System/3mea-di5p (28.05.2021)
- http://gis.humangeo.su.se/qgistutor/www.qgistutorials.com/en/docs/3/basic_network_analysis.html (20.05.2021)

GROUP MEMBERS;

- Emre GÖKBULUT – 21632624
- İrem BAKIR – 21732881
- Oktay ÇAKMAK – 21732991
- Berkay AKINCI – 21967277


THANK YOU
FOR
LISTENING !!!