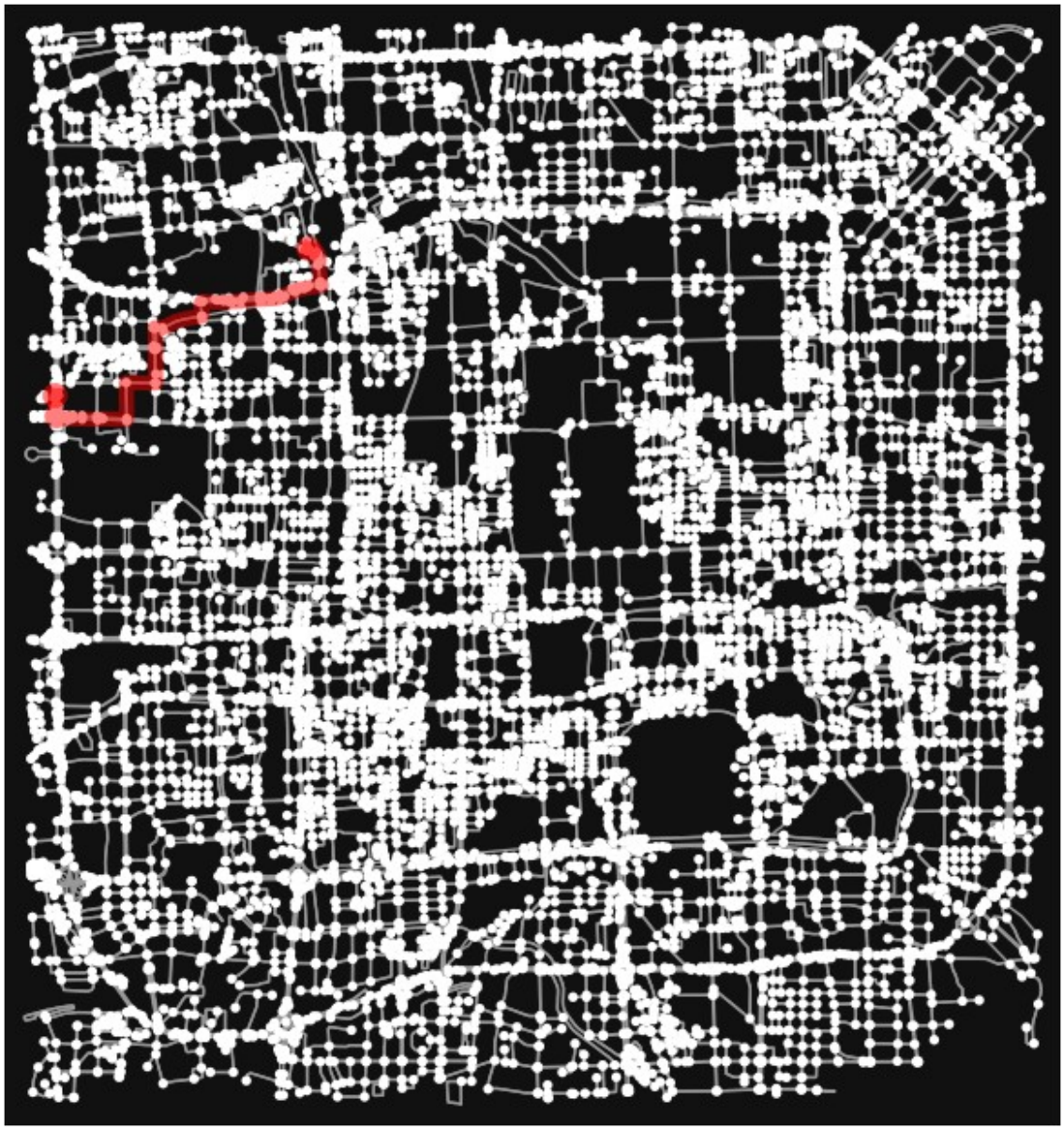
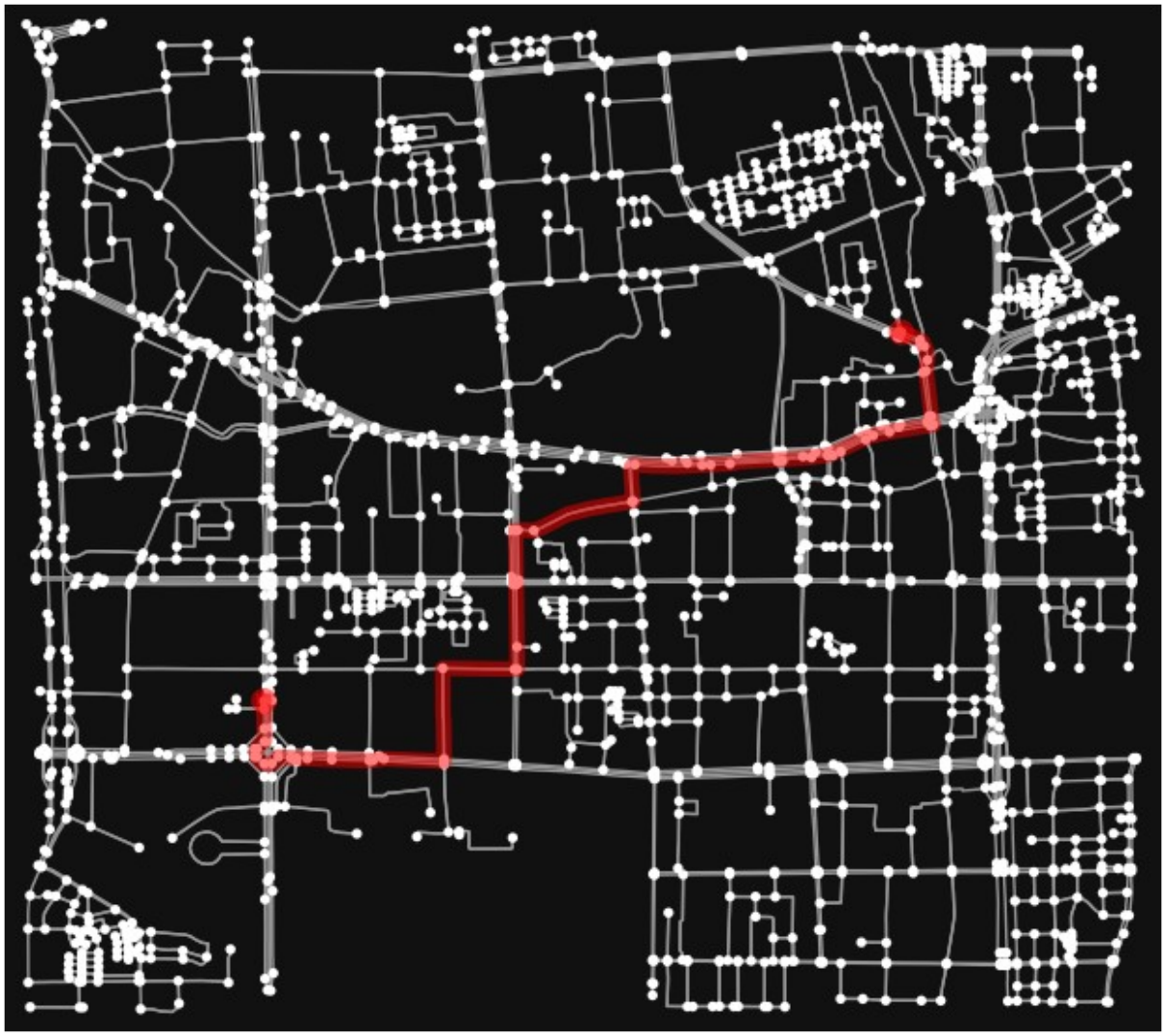
```python
# Importing libraries
import networkx as nx # Library for containing graphs
import osmnx as ox # Library for open street maps
from random import randint # Random
import matplotlib.pyplot as plt # Plots

# Study Area
min_lat = 39.84
min_lon = 116.30
max_lat = 39.97
max_lon = 116.46

# Getting study area graph (with roads for cars)
G = ox.graph_from_bbox(bbox=(min_lat, max_lat, min_lon, max_lon),
network_type='drive')

# Two random points on the map which I will use to calculate route
point1 = list(G.nodes)[randint(0, len(G.nodes)-1)]
point2 = list(G.nodes)[randint(0, len(G.nodes)-1)]

# Calculating shortest path between two points with dijkstra algorithm
shortest_path = nx.shortest_path(G, target=point1, source=point2,
method="dijkstra", weight='length')
# Printing nodes in the shortest path
print("Shortest path:", shortest_path)
```

```
Shortest path: [1361129395, 8727881497, 604818952, 324263527,
324263455, 324264189, 324264264, 324264294, 324263371, 604820919,
9695437195, 1930829008, 1930829004, 324264150, 1361129180, 1361129190,
9776323763, 9776323762, 754515896, 321939060, 321939059, 11529936534,
1772777223, 331148410, 321938953, 321939013, 1573977228, 321939012,
10032701333, 321939008, 10032701372, 1253534960, 1292883489,
1415627433, 656086941, 656086877, 656086946, 9949183417, 9949178516,
9949183421, 9385323973, 265710694, 290598250, 2108372962, 9385323981]
```

```python
# Figure 1: Road network plot using OSMnx
fig, ax = ox.plot_graph_route(G, shortest_path, route_linewidth=6,
route_color='r')
```

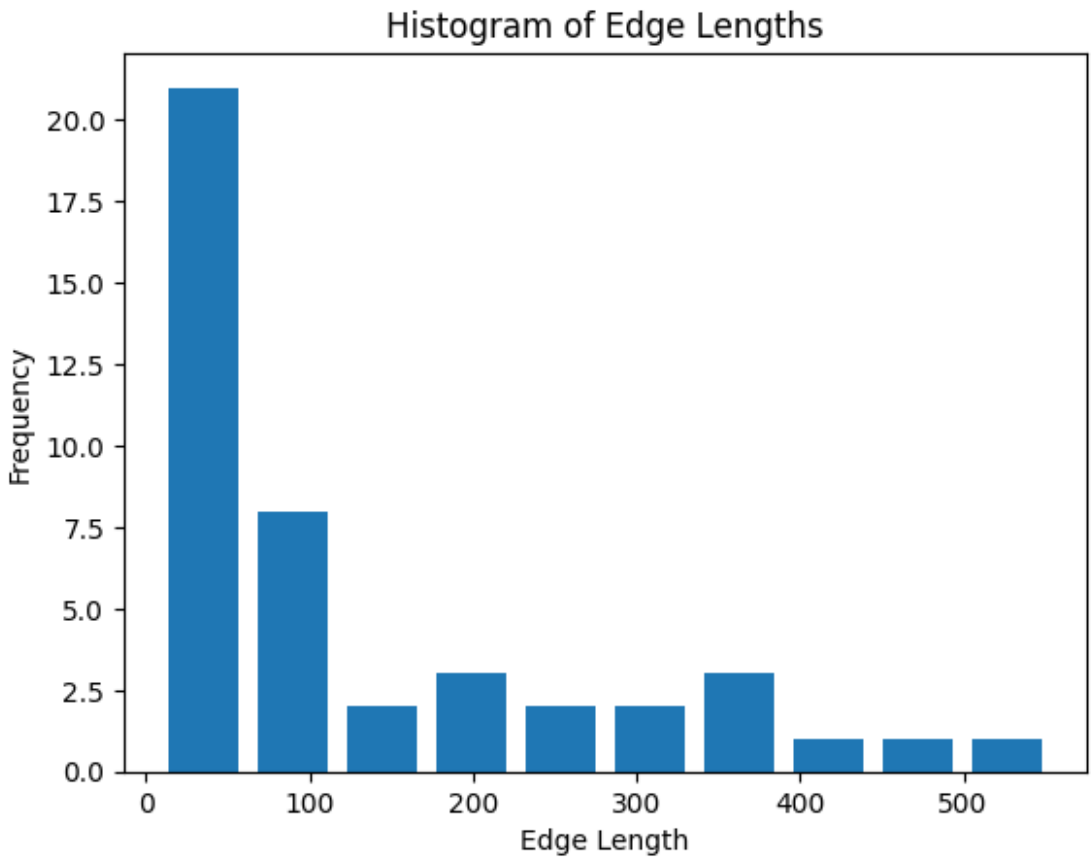


```
# Figure 2: Scaled map of our route
list_shortest_path = list(shortest_path)
G_path = G.subgraph(shortest_path)
# Creating new mini graph (using x and y from start and end of the
route. also adding 0.015 to have more place)
G_mini = ox.graph_from_bbox(bbox=(G.nodes[list_shortest_path[0]]["y"]-
0.015, G.nodes[list_shortest_path[-1]]["y"]+0.015,
G.nodes[list_shortest_path[0]]["x"]-0.015,
G.nodes[list_shortest_path[-1]]["x"]+0.015), network_type='drive')
fig2, ax2 = ox.plot_graph_route(G_mini, shortest_path,
route_linewidth=6, route_color='r')
```

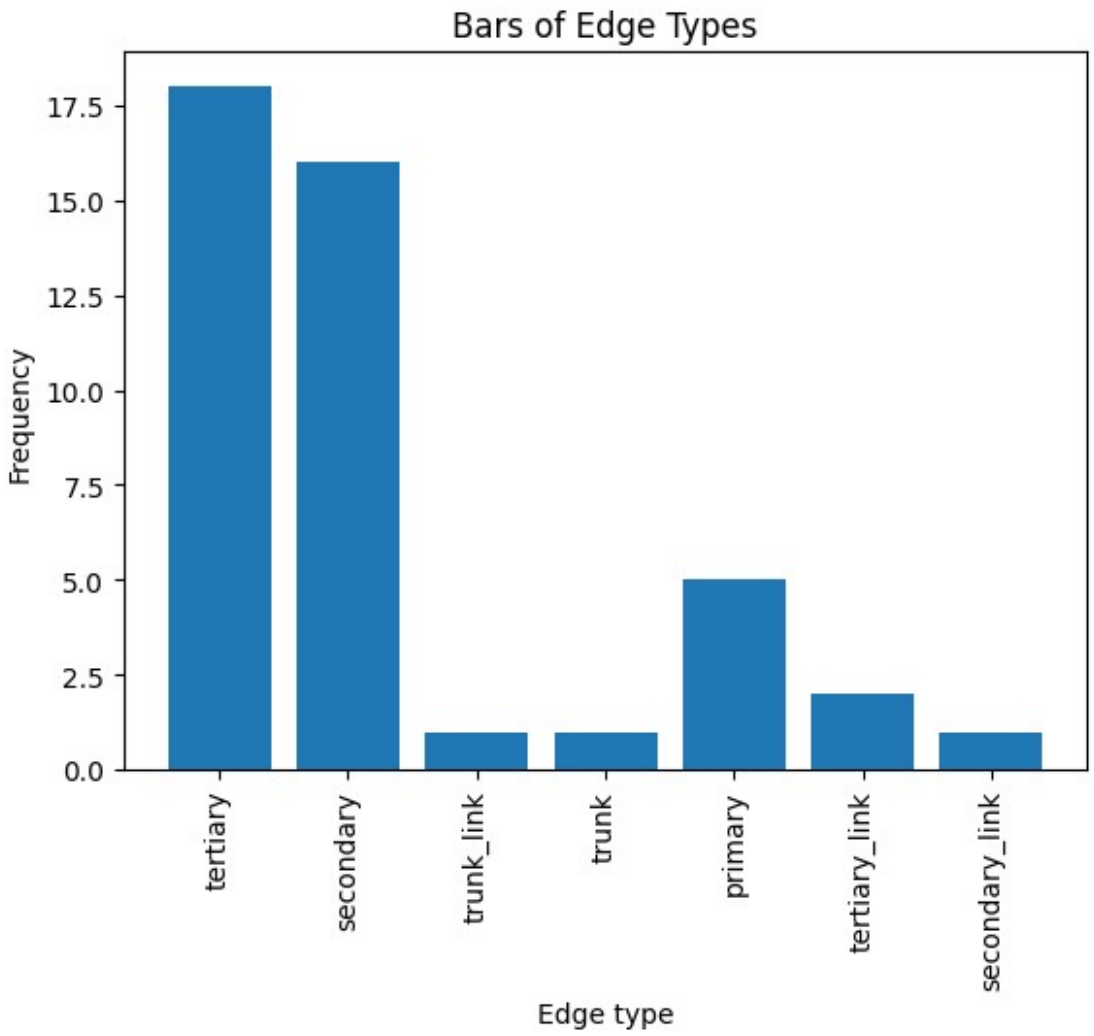


```
# Figure 3: Histogram of edge lengths
# Calculate the lengths of edges
edge_lengths = [G.get_edge_data(list_shortest_path[i],
list_shortest_path[i + 1])[0]['length'] for i in
range(len(list_shortest_path) - 1)]
# Create a histogram of edge lengths
plt.hist(edge_lengths, rwidth=0.8, histtype='bar')
plt.xlabel('Edge Length')
plt.ylabel('Frequency')
plt.title('Histogram of Edge Lengths')
```

```
Text(0.5, 1.0, 'Histogram of Edge Lengths')
```




```
# Figure 4: Bars of different types of roads
edge_types = [G.get_edge_data(list_shortest_path[i],
list_shortest_path[i + 1])[0]['highway'] for i in
range(len(list_shortest_path) - 1)]
for i in range(len(edge_types)):
    if type(edge_types[i]) != str:
        edge_types[i] = ''
edge_types = list(filter(None, edge_types))
# Create bars
type_counts = {}
for t in edge_types:
    type_counts[t] = type_counts.get(t, 0) + 1

type_labels = list(type_counts.keys())
frequencies = list(type_counts.values())
plt.bar(type_labels, frequencies)
plt.xticks(rotation='vertical')
plt.xlabel('Edge type')
plt.ylabel('Frequency')
plt.title('Bars of Edge Types')

Text(0.5, 1.0, 'Bars of Edge Types')
```




```
# Saving graph and shortest path for QGIS
ox.save_graph_geopackage(G, 'map')
ox.save_graph_geopackage(G_path, 'shortest_path')
```

Presenting graph in QGIS

title