

Import Libraries

```
In [26]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from geopy.distance import geodesic
from IPython.display import Image
```

```
In [27]: from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

Parsing Location Coordinates

```
In [28]: ride = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/rideshare_kaggle.csv")
# ride = pd.read_csv('rideshare_kaggle.csv')

x = ride[["latitude", "longitude"]]
```

```
In [29]: coords = {
    'Haymarket Square': [42.364,-71.060],
    'Back Bay' : [42.3503,-71.0810],
    'North End': [42.3647,-71.0542],
    'North Station': [42.3661,-71.0631],
    'Beacon Hill': [42.3588,-71.0707],
    'Boston University': [42.3505,-71.1054],
    'Fenway': [42.3505,-71.1054],
    'South Station': [42.3519,-71.0551],
    'Theatre District': [42.352,-71.065],
    'West End': [42.3661,-71.0631],
    'Financial District' : [42.3559,-71.0550],
    'Northeastern University': [42.3398,-71.0892]
}
```

```
def parse_lat(document):

    ret = [0,0]
    if document:
        lat = coords[document][0]
    return lat
```

```
def parse_long(document):

    ret = [0,0]
    if document:
        lon = coords[document][1]
    return lon
```

```
In [30]: s_coords1 = ride["source"].apply(parse_lat)
s_coords2 = ride["source"].apply(parse_long)
d_coords1 = ride["destination"].apply(parse_lat)
d_coords2 = ride["destination"].apply(parse_long)
```

```
l1 = s_coords1.append(d_coords1, ignore_index = True)
l2 = s_coords2.append(d_coords2, ignore_index = True)
```

<ipython-input-30-a8d0bef7471d>:6: FutureWarning: The series.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

```
l1 = s_coords1.append(d_coords1, ignore_index = True)
```

<ipython-input-30-a8d0bef7471d>:7: FutureWarning: The series.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

```
l2 = s_coords2.append(d_coords2, ignore_index = True)
```

```
In [31]: x = pd.concat([l1,l2],axis=1)
print(x)
```

```
      0      1
0    42.3640 -71.0600
1    42.3640 -71.0600
2    42.3640 -71.0600
3    42.3640 -71.0600
4    42.3640 -71.0600
...    ...    ...
1386137 42.3647 -71.0542
1386138 42.3647 -71.0542
1386139 42.3647 -71.0542
1386140 42.3647 -71.0542
1386141 42.3647 -71.0542
```

```
[1386142 rows x 2 columns]
```

```
In [32]: pip install -U googlemaps
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) http://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-python.pkg.dev/colab-wheels/public/simple/)
Requirement already satisfied: googlemaps in /usr/local/lib/python3.9/dist-packages (4.10.0)
Requirement already satisfied: requests<3.0,>=2.20.0 in /usr/local/lib/python3.9/dist-packages (from googlemaps) (2.27.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests<3.0,>=2.20.0->googlemaps) (3.4)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests<3.0,>=2.20.0->googlemaps) (2.0.12)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packages (from requests<3.0,>=2.20.0->googlemaps) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-packages (from requests<3.0,>=2.20.0->googlemaps) (1.26.15)
```

```
In [33]: import googlemaps
```

```
gmaps = googlemaps.Client(key='AIzaSyB-Dvs504EXQNMt1L-S79Yil2I58dCeJLQ')
```

Plotting Hubs using Heatmap

```
In [34]: l1 = s_coords1.append(d_coords1, ignore_index = True)
l2 = s_coords2.append(d_coords2, ignore_index = True)

import folium
from folium.plugins import HeatMap
hmap = folium.Map(location=[42.35, -71.07], zoom_start=13, title="Most Common

hm_wide = HeatMap( list(zip(l1,l2)),
                    min_opacity=0.5,
                    radius=25, blur=20,
                    max_zoom=1,
                    )

hmap.add_child(hm_wide)
```

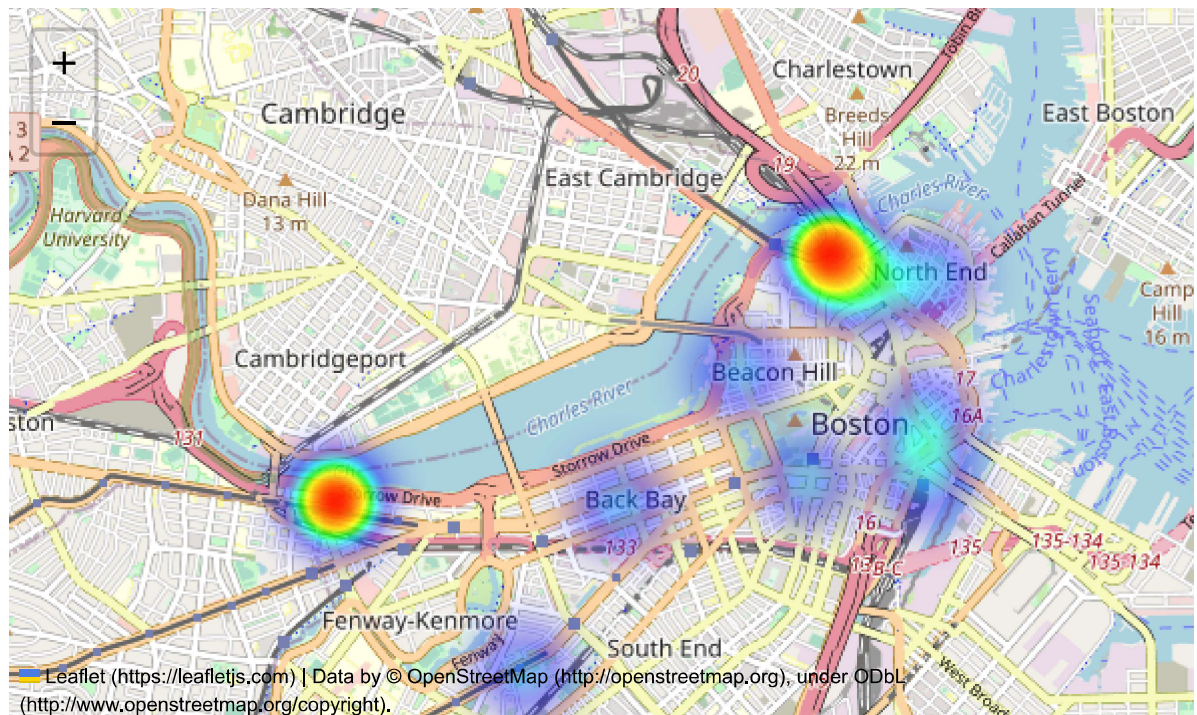
<ipython-input-34-5e371f4aa757>:1: FutureWarning: The series.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

```
l1 = s_coords1.append(d_coords1, ignore_index = True)
```

<ipython-input-34-5e371f4aa757>:2: FutureWarning: The series.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

```
l2 = s_coords2.append(d_coords2, ignore_index = True)
```

Out[34]:



Example Route Through all Hubs

```

In [35]: waypoints = [
    'Back Bay, Boston',
    'North End, Boston',
    'North Station, Boston',
    'Beacon Hill, Boston',
    'Boston University, Boston',
    'Fenway, Boston',
    'South Station, Boston',
    'Theatre District, Boston',
    'West End, Boston',
    'Financial District, Boston',
    ]

results = gmaps.directions(origin = 'Haymarket Square, Boston',
                            destination = 'Northeastern University, Boston',
                            waypoints = waypoints,
                            optimize_waypoints = True)

for i, leg in enumerate(results[0]["legs"]):
    print("Stop:" + str(i),
          leg["start_address"],
          "==> ",
          leg["end_address"],
          "distance: ",
          leg["distance"]["value"],
          "traveling Time: ",
          leg["duration"]["value"]
    )

```

```

Stop:0 Boston, MA, USA ==> West End, Boston, MA, USA distance: 843 traveling Time: 286
Stop:1 West End, Boston, MA, USA ==> North Station, 135 Causeway St, Boston, MA 02114, USA distance: 88 traveling Time: 33
Stop:2 North Station, 135 Causeway St, Boston, MA 02114, USA ==> North End, Boston, MA, USA distance: 1091 traveling Time: 522
Stop:3 North End, Boston, MA, USA ==> Financial District, Boston, MA, USA distance: 1565 traveling Time: 494
Stop:4 Financial District, Boston, MA, USA ==> South Station, 700 Atlantic Ave, Boston, MA 02110, USA distance: 543 traveling Time: 157
Stop:5 South Station, 700 Atlantic Ave, Boston, MA 02110, USA ==> Boston Theater District, Boston, MA, USA distance: 1948 traveling Time: 636
Stop:6 Boston Theater District, Boston, MA, USA ==> Beacon Hill, Boston, MA, USA distance: 1564 traveling Time: 535
Stop:7 Beacon Hill, Boston, MA, USA ==> Back Bay, Boston, MA, USA distance: 1453 traveling Time: 346
Stop:8 Back Bay, Boston, MA, USA ==> Boston, MA 02215, USA distance: 2044 traveling Time: 443
Stop:9 Boston, MA 02215, USA ==> Fenway-Kenmore, Boston, MA, USA distance: 2970 traveling Time: 326
Stop:10 Fenway-Kenmore, Boston, MA, USA ==> 360 Huntington Ave, Boston, MA 02115, USA distance: 542 traveling Time: 97

```



```

In [36]: marker_points = []
         waypoints = []

         #extract the Location points from the previous directions function

         for leg in results[0]["legs"]:
             leg_start_loc = leg["start_location"]
             marker_points.append(f'{leg_start_loc["lat"]},{leg_start_loc["lng"]}')
             for step in leg["steps"]:
                 end_loc = step["end_location"]
                 waypoints.append(f'{end_loc["lat"]},{end_loc["lng"]}')
             last_stop = results[0]["legs"][-1]["end_location"]
             marker_points.append(f'{last_stop["lat"]},{last_stop["lng"]}')

         markers = [ "color:blue|size:mid|label:" + chr(65+i) + "|"
                     + r for i, r in enumerate(marker_points)]
         result_map = gmaps.static_map(
             center = waypoints[0],
             scale=2,
             zoom=13,
             size=[500, 300],
             format="jpg",
             maptype="roadmap",
             markers=markers,
             path="color:0x0000ff|weight:2|" + "|".join(waypoints))

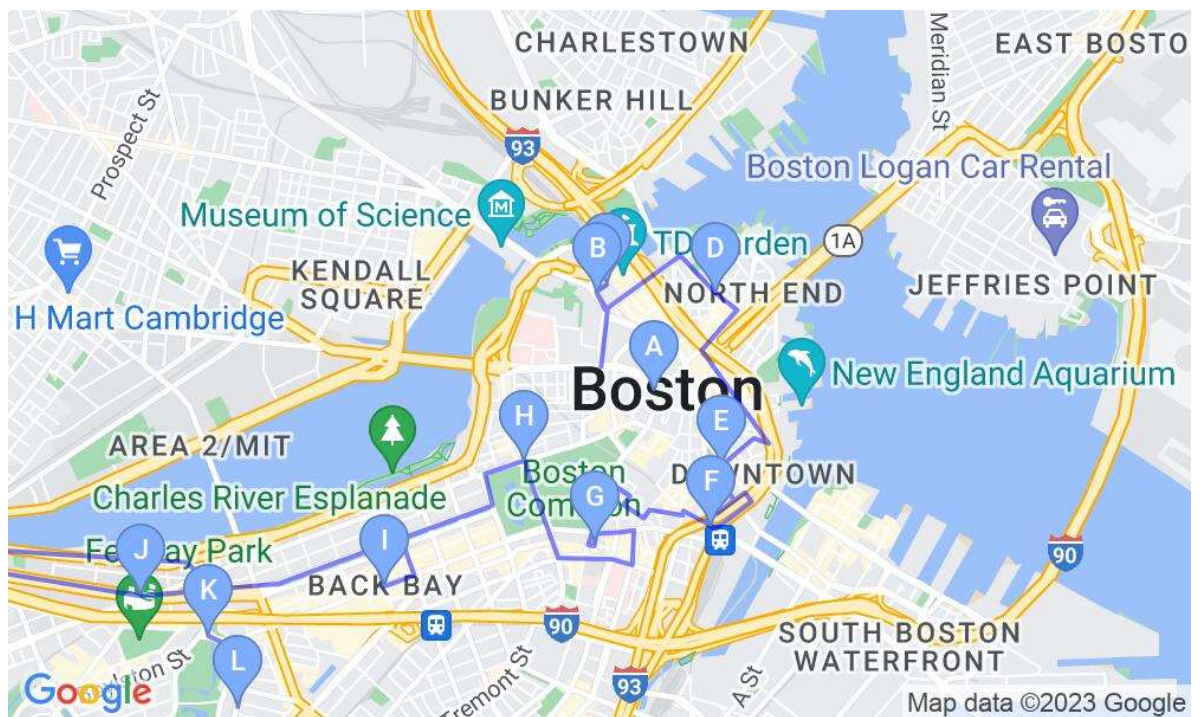
```

```

In [37]: with open('driving_route_map.jpg', 'wb') as img:
         for chunk in result_map:
             img.write(chunk)

         i = Image('driving_route_map.jpg')
         display(i)

```



Simple Route Between 2 Hubs

```
In [38]: locations = [  
    'Haymarket Square, Boston',  
    'Back Bay, Boston',  
    'North End, Boston',  
    'North Station, Boston',  
    'Beacon Hill, Boston',  
    'Boston University, Boston',  
    'Fenway, Boston',  
    'South Station, Boston',  
    'Theatre District, Boston',  
    'West End, Boston',  
    'Financial District, Boston',  
    'Northeastern University, Boston']  
  
location_geocodes = {}  
  
for loc in locations:  
    location_geocodes[loc] = gmaps.geocode(loc)
```



```

In [39]: print('1. Haymarket Square')
print('2. Back Bay')
print('3. North End')
print('4. North Station')
print('5. Beacon Hill')
print('6. Boston University')
print('7. Fenway')
print('8. South Station')
print('9. Theatre District')
print('10. West End')
print('11. Financial District')
print('12. Northeastern University')

print("-----")
s = input("Select Source Option: ")
d = input("Select Destination Option: ")
source = locations[int(s)-1]
destination = locations[int(d)-1]
print("-----")
print('Selected Source: ',source)
print('Selected Destination: ',destination)

results = gmaps.directions(origin = source,
                             destination = destination,
                             optimize_waypoints = True)

marker_points = []
waypoints = []

#extract the location points from the previous directions function

for leg in results[0]["legs"]:
    leg_start_loc = leg["start_location"]
    marker_points.append(f'{leg_start_loc["lat"]},{leg_start_loc["lng"]}')
    for step in leg["steps"]:
        end_loc = step["end_location"]
        waypoints.append(f'{end_loc["lat"]},{end_loc["lng"]}')
last_stop = results[0]["legs"][-1]["end_location"]
marker_points.append(f'{last_stop["lat"]},{last_stop["lng"]}')

markers = [ "color:blue|size:mid|label:" + chr(65+i) + "|"
            + r for i, r in enumerate(marker_points)]

result_map2 = gmaps.static_map(
    center = waypoints[0],
    scale=2,
    zoom=13,
    size=[500, 300],
    format="jpg",
    maptype="roadmap",
    markers=markers,
    path="color:0x0000ff|weight:2|" + "|".join(waypoints))

with open('driving_route_map2.jpg', 'wb') as img:
    for chunk in result_map2:
        img.write(chunk)

```

```
i = Image('driving_route_map2.jpg')
display(i)
```

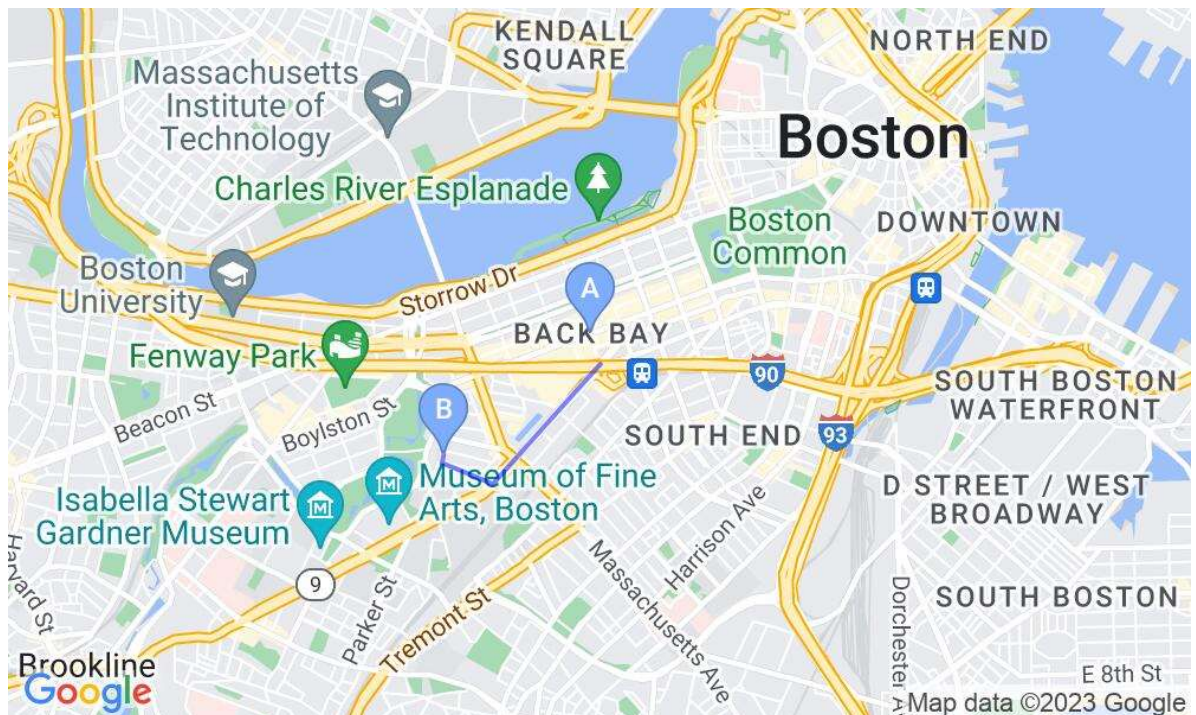
1. Haymarket Square
2. Back Bay
3. North End
4. North Station
5. Beacon Hill
6. Boston University
7. Fenway
8. South Station
9. Theatre District
10. West End
11. Financial District
12. Northeastern University

Select Source Option: 2

Select Destination Option: 12

Selected Source: Back Bay, Boston

Selected Destination: Northeastern University, Boston



Routing Any Possible Rideshare Routes Through Hubs

In [40]:

```
def route():

    print("-----")
    s = input("Select Source Address: ")
    d = input("Select Destination Address: ")
    print("-----")

    # s = "Nickerson Field"
    # d = "John F. Kennedy Presidential Library and Museum"

    source = gmaps.geocode(s)
    destination = gmaps.geocode(d)
    print(source[0]["formatted_address"])
    print(source[0]["geometry"]["location"]["lat"])
    print(source[0]["geometry"]["location"]["lng"])
    print()
    print(destination[0]["formatted_address"])
    print(destination[0]["geometry"]["location"]["lat"])
    print(destination[0]["geometry"]["location"]["lng"])
    print()

    max_s_dist = float('inf')
    max_d_dist = float('inf')
    closest_source_hub = locations[0]
    closest_dest_hub = locations[0]

    for loc, loc_geo in location_geocodes.items():
        curr_dist1 = gmaps.distance_matrix(origins=source[0]['formatted_address'],
                                           destinations=loc_geo[0]["formatted_address"])[0][0]
        curr_dist2 = gmaps.distance_matrix(origins=loc_geo[0]['formatted_address'],
                                           destinations=destination[0]["formatted_address"])[0][0]

        if curr_dist1 < max_s_dist:
            closest_source_hub = loc
            max_s_dist = curr_dist1
        if curr_dist2 < max_d_dist:
            closest_dest_hub = loc
            max_d_dist = curr_dist2

    print("Closest hub to source = ", closest_source_hub)
    print("Closest hub to destination = ", closest_dest_hub)

    waypoints = [source[0]['formatted_address'], closest_source_hub, closest_dest_

    results = gmaps.directions(origin = source[0]['formatted_address'],
                               destination = destination[0]['formatted_address'],
                               waypoints = waypoints,
                               optimize_waypoints = True)

    w = waypoints
    marker_points = []
    waypoints = []
```

#extract the Location points from the previous directions function

```
for leg in results[0]["legs"]:
    leg_start_loc = leg["start_location"]
    marker_points.append(f'{leg_start_loc["lat"]},{leg_start_loc["lng"]}')
    for step in leg["steps"]:
        end_loc = step["end_location"]
        waypoints.append(f'{end_loc["lat"]},{end_loc["lng"]}')
last_stop = results[0]["legs"][-1]["end_location"]
marker_points.append(f'{last_stop["lat"]},{last_stop["lng"]}')

markers = [ "color:blue|size:mid|label:" + chr(65+i) + "|"
            + r for i, r in enumerate(w)]

result_map = gmaps.static_map(
    scale=2,
    zoom=12,
    size=[500, 300],
    format="jpg",
    maptype="roadmap",
    markers=markers,
    path="color:0x0000ff|weight:2|" + "|".join(waypoints))

with open('driving_route_map2.jpg', 'wb') as img:
    for chunk in result_map:
        img.write(chunk)

i = Image('driving_route_map2.jpg')
display(i)
```

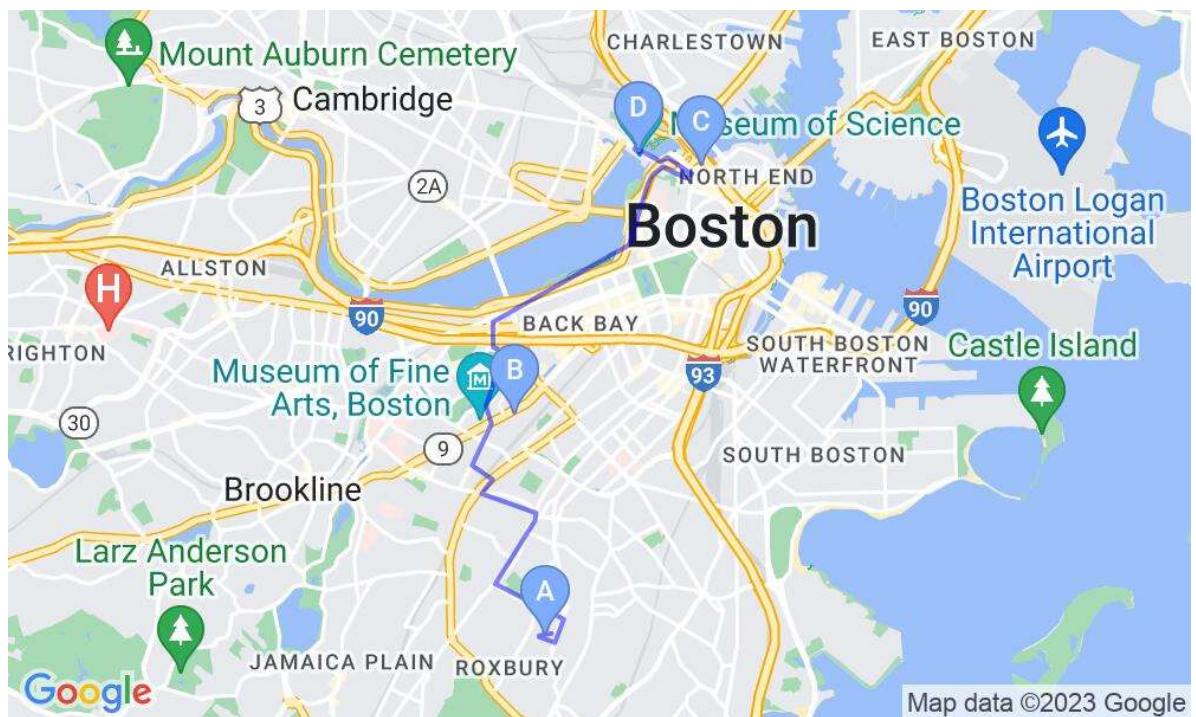
```
In [41]: route()
```

```
-----  
Select Source Address: boston latin academy  
Select Destination Address: museum of science  
-----
```

```
205 Townsend St, Boston, MA 02121, USA  
42.316119  
-71.0845953
```

```
Museum Of Science Driveway, Boston, MA 02114, USA  
42.367714  
-71.0709771
```

```
Closest hub to source = Northeastern University, Boston  
Closest hub to destination = North Station, Boston
```




```
In [44]: route()
```

Select Source Address: MIT

Select Destination Address: boston university

77 Massachusetts Ave, Cambridge, MA 02139, USA

42.360091

-71.09416

Boston, MA 02215, USA

42.3504997

-71.1053991

Closest hub to source = Fenway, Boston

Closest hub to destination = Boston University, Boston



In [42]: route()

Select Source Address: MIT

Select Destination Address: BU medical campus

77 Massachusetts Ave, Cambridge, MA 02139, USA

42.360091

-71.09416

715 Albany St #437, Boston, MA 02118, USA

42.335579

-71.0711474

Closest hub to source = Fenway, Boston

Closest hub to destination = Northeastern University, Boston



In []: