

ASSIGNMENT 4 : Tracking the NASA Satellite

For this assignment, we will be working on streaming data which is satellite location coordinates being provided by NASA.

The data is in the following format:

```
{"timestamp": 1667492679, "iss_position": {"longitude": "-56.8155", "latitude": "-42.5979"}, "message": "success"}
```

Data can be accessed from <http://api.open-notify.org/iss-now.json>

The link can be thought of as the producer as it will do producer job.

Write the consumer to connect and fetch data. In the consumer, you have to write the code which takes the data from producer and uses the location coordinates to plot the satellite location on the world map.

The streaming interval should be 5 seconds. The producer should run for an hour. The graph will show the satellite tracking for 1 hr.

P.S.: Keeping in mind that satellite travels very fast, the graph will have more than half of map covered with satellite track

```
In [1]: import requests
        from mpl_toolkits.basemap import Basemap
        import matplotlib.pyplot as plt
        import time

        # Set up the basemap
        map = Basemap(projection='merc', resolution='l', area_thresh=1000.0,
                      llcrnrlon=-180, llcrnrlat=-80, urcnrlon=180, urcnrlat=80)

        # Set up the figure
        fig = plt.figure(figsize=(10, 8))
        ax = fig.add_subplot(111)

        # Initialize the coordinates list
        lons, lats = [], []

        # Set up the API endpoint
```

```
url = 'http://api.open-notify.org/iss-now.json'

# Set the start time
start_time = time.time()

# Run the producer for an hour
while time.time() - start_time < 3600:
    # Fetch the data
    response = requests.get(url).json()
    timestamp = response['timestamp']
    longitude = float(response['iss_position']['longitude'])
    latitude = float(response['iss_position']['latitude'])

    # Add the coordinates to the list
    lons.append(longitude)
    lats.append(latitude)

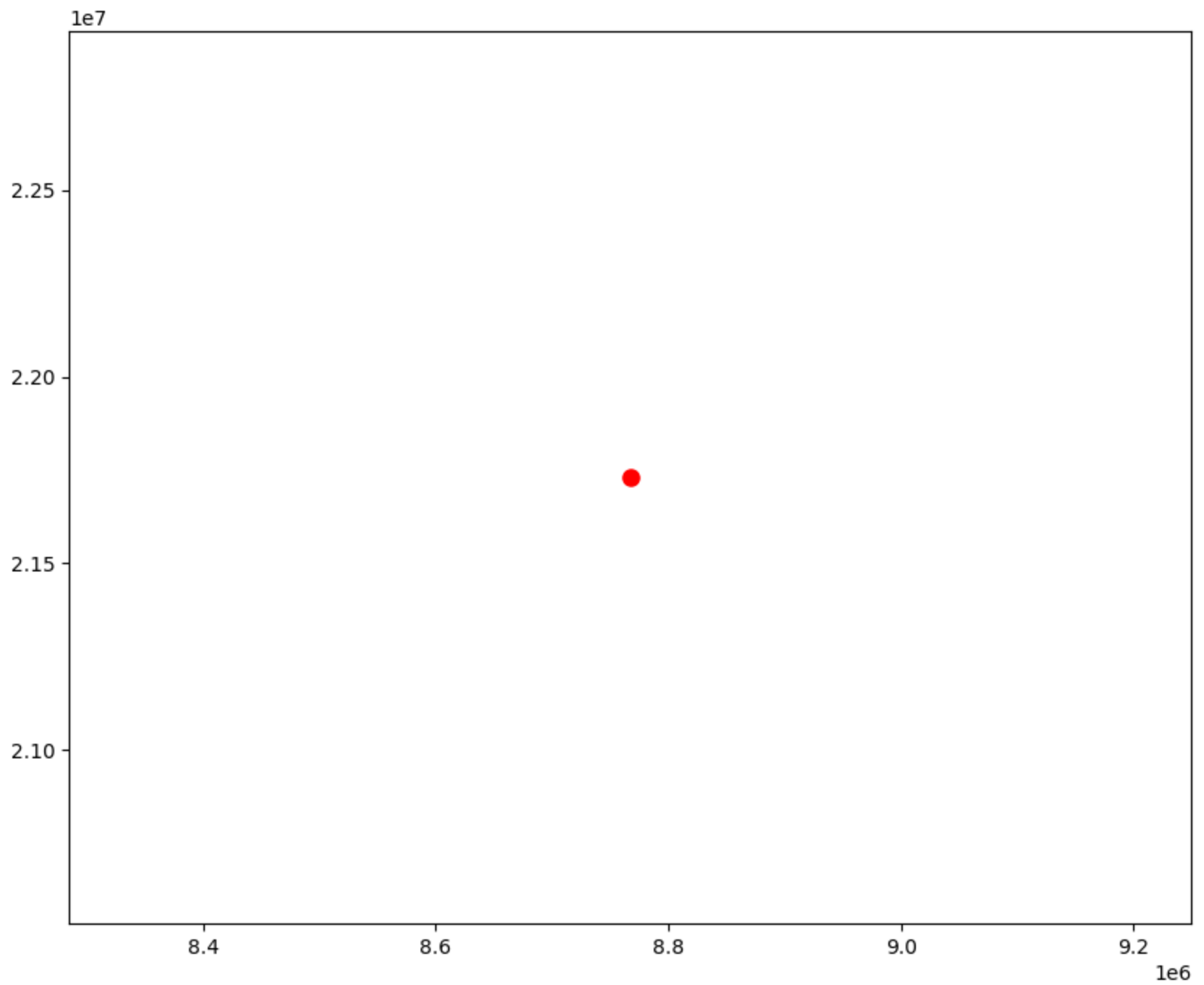
    # Plot the current location of the satellite
    x, y = map(longitude, latitude)
    ax.plot(x, y, 'ro', markersize=8)

    # Refresh the plot
    plt.draw()
    plt.pause(0.001)

    # Wait for 5 seconds
    time.sleep(5)

# Plot the satellite track on the world map
map.drawcoastlines()
map.drawcountries()
map.drawmapboundary(fill_color='aqua')
map.fillcontinents(color='coral', lake_color='aqua')
map.drawmeridians(range(-180, 180, 60), labels=[False, False, False, True])
map.drawparallels(range(-90, 90, 30), labels=[True, False, False, False])
x, y = map(lons, lats)
ax.plot(x, y, 'b-', linewidth=2)

# Show the plot
plt.show()
```



<Figure size 640x480 with 0 Axes>

[illegible]

file:///C:/Users/chand/Downloads/ChandradeepReddy_HW4.html

[illegible]

[illegible]

file:///C:/Users/chand/Downloads/ChandradeepReddy_HW4.html

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>
<Figure size 640x480 with 0 Axes>

