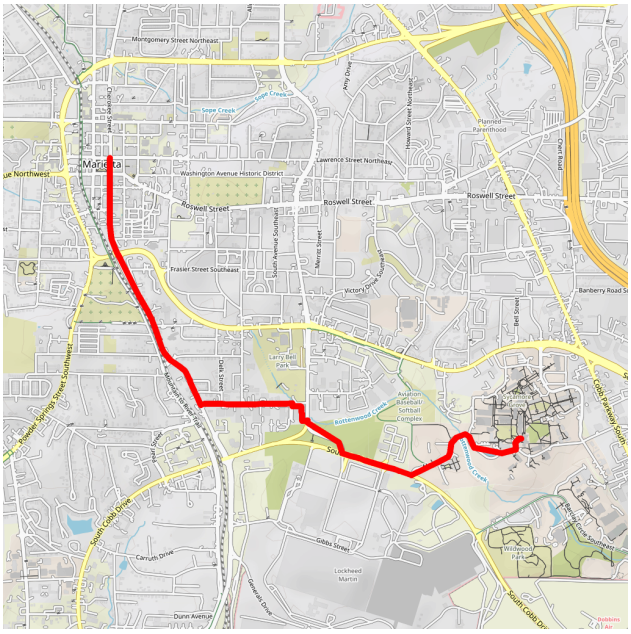
Homework 1 (Due January 26)

CS7253: Graph Algorithms
Kennesaw State University
Spring 2024

In this project, we will implement Google Maps (sort of). In particular, we will implement Dijkstra's algorithm, and run it on a map of Marietta. Our goal is to find the shortest path (in meters) between two given places in Marietta, for example, from the J Building at KSU to Sweetreats at Marietta Square:



Skeleton code is provided to help you get started. It includes implementations of the following classes:

- a `Graph` class, in the file `graph.py`, is a simple implementation of an undirected graph, as a list of (weighted) edges. It also maintains a list of neighbors for every node in the graph, and includes a function to compute the length of a given path.
- a `Map` class, in the file `maps.py`, which maintains the map of Marietta, as a graph. Each node is an intersection, and each edge is a street. Each node has a coordinate as a $(latitude, longitude)$ pair.

In this project, you are to implement the function

`shortest_path(graph,s,t)`

defined in the file `dijkstra.py`. This function should return the shortest path from node `s` to node t in a given graph `graph`. A path is represented as a list of edges, where an edge is a pair of nodes. A node is represented by an integer id. You should include any helper functions in `dijkstra.py`, as needed.

The file `main.py` is an example program that includes two tests. The second test corresponds to the example we covered in class. The first test corresponds to a map of Marietta, and requests a path from the J Building at KSU to Sweetreats at Marietta Square. The expected shortest path has a length of 4,337 meters (or 2.69488 miles). (Compare this to the route suggested by google maps).

Visualization: The file `maps.py` includes a function `draw_path` that can draw a given path on a map of Marietta, like the one above. To use this function, the Python module `staticmap` must be installed.

Turn in: your modified version of `dijkstra.py` onto the course website under `Assignments` and `Homework 1`. Turn in `dijkstra.py` **only**. Your implementation of the `shortest_path` function should not require any changes outside of `dijkstra.py`. Assignments are due Friday, January 26 by 11:59pm. Please start early in case you encounter any unexpected difficulties.

Testing and Grading: The homework assignment includes tests in the file `tests.py`. These are `pytest` tests, and should not be modified. If you want to run these tests, you can install `pytest` and then run the command:

```
pytest tests.py
```

to run all tests. If you code is implemented correctly, you should see 5 passing tests:

```
============================= test session starts ==============================
platform linux -- Python 3.11.6, pytest-7.4.4, pluggy-1.3.0
rootdir: /kennesaw/cs7253/spring24/hw1/
plugins: cov-4.1.0
collected 5 items

tests.py .....                                                          [100%]

============================== 5 passed in 0.09s ===============================
```

Your homework will be graded based on a suite of *private* tests, which may include the public tests provided with the assignment. If your script passes all private tests, your score will be 100 our of 100. If your script does not run, you will fail all tests, and receive a score of zero. Do not import any modules beyond those that are already imported for you in the homework. Missing imports will cause tests to fail. New modules will not be installed on the testing system. Any identical code submissions will also receive a score of zero.

Included files:

- `homework01.pdf`: this document
- `dijkstra.py`: the script you implement and turn in
- `graph.py`: the script containing the `graph` class
- `main.py`: an example program that you can run
- `maps.py`: the script containing the `Map` class
- `tests.py`: the script containing the `pytest` tests
- `maps/`: a directory that contains the Marietta map and image

Hint: Use the debugger.

Notes: The map of Marietta was originally downloaded from the following address:

`https://www.openstreetmap.org/export#map=15/33.9470/-84.5259`

The original map downloaded from `openstreetmap` is an .xml file, which was processed usingthe Java library `graphhopper` for reading maps and for projecting GPS coordinates onto a map.