Graph Lab

Header information:

- Author #1: Krish Patel (patek74@mcmaster.ca)
- Author #2: Harsh Chinjer (chinjerh@mcmaster.ca)
- Gitlab URL: https://gitlab.cas.mcmaster.ca/patek74/l1-graph-lab
- Avenue to Learn group name: Group 21

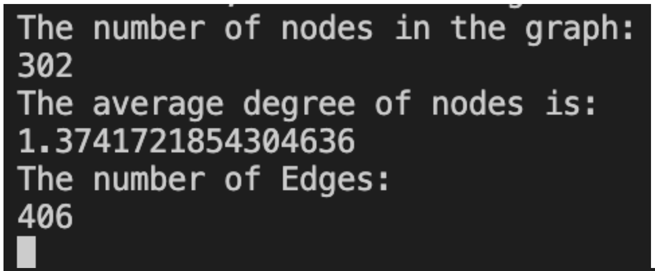
Week 1

Metrics Computed:

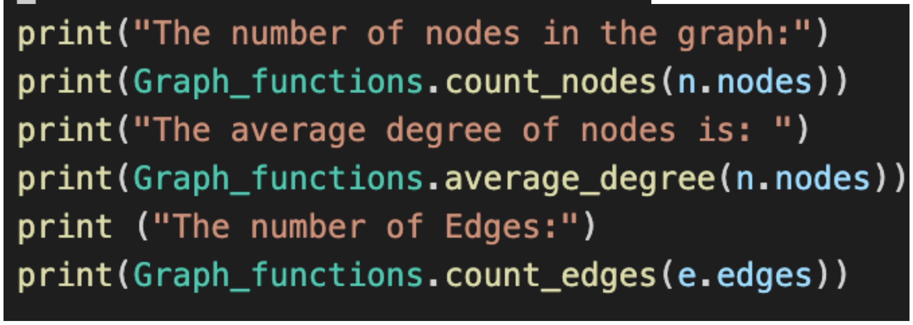
Number of nodes: 302

The average degree: 1.37

The number of edges: 406



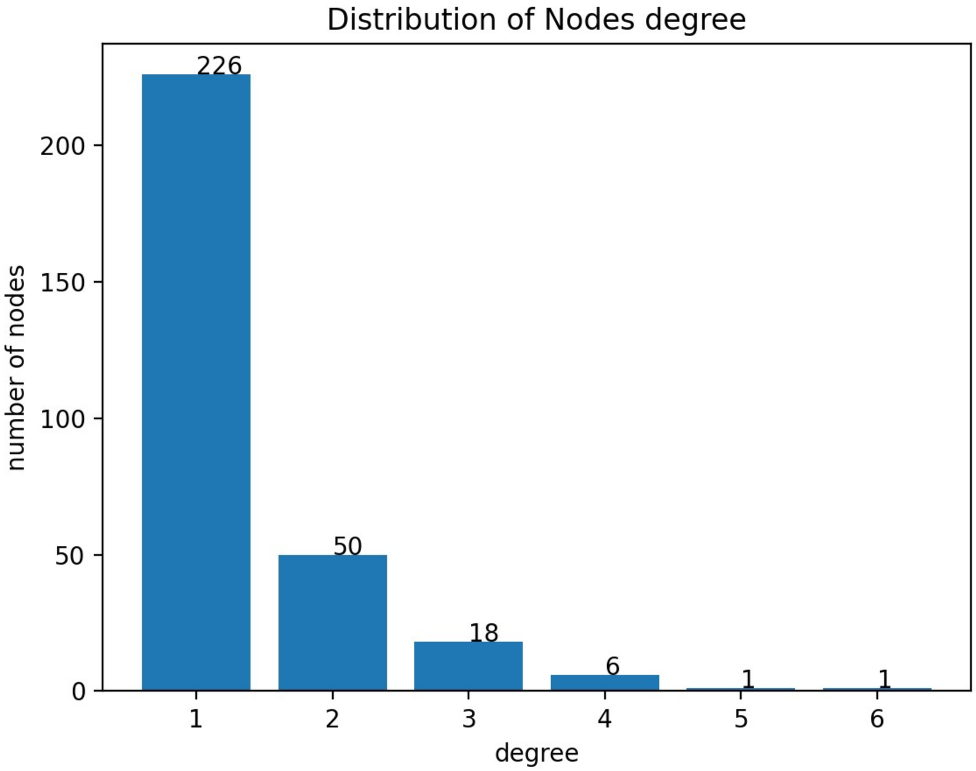
```
The number of nodes in the graph:
302
The average degree of nodes is:
1.3741721854304636
The number of Edges:
406
```



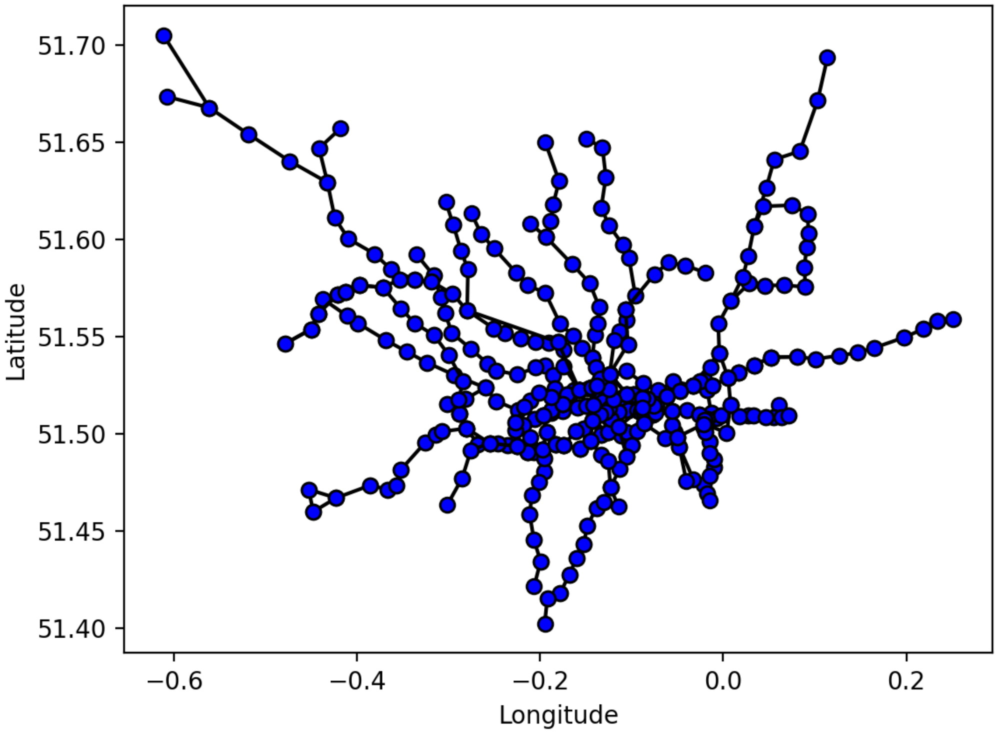
```
print("The number of nodes in the graph:")
print(Graph_functions.count_nodes(n.nodes))
print("The average degree of nodes is: ")
print(Graph_functions.average_degree(n.nodes))
print ("The number of Edges:")
print(Graph_functions.count_edges(e.edges))
```

```
def count_nodes(nodes): #returns node count
    return len(nodes)

def count_edges(edges): #returns edge count
     return len(edges)

def average_degree(nodes):
    total=0
    for node in nodes: #degree of all nodes
        total+=node.degree
    return (total/len(nodes))
```

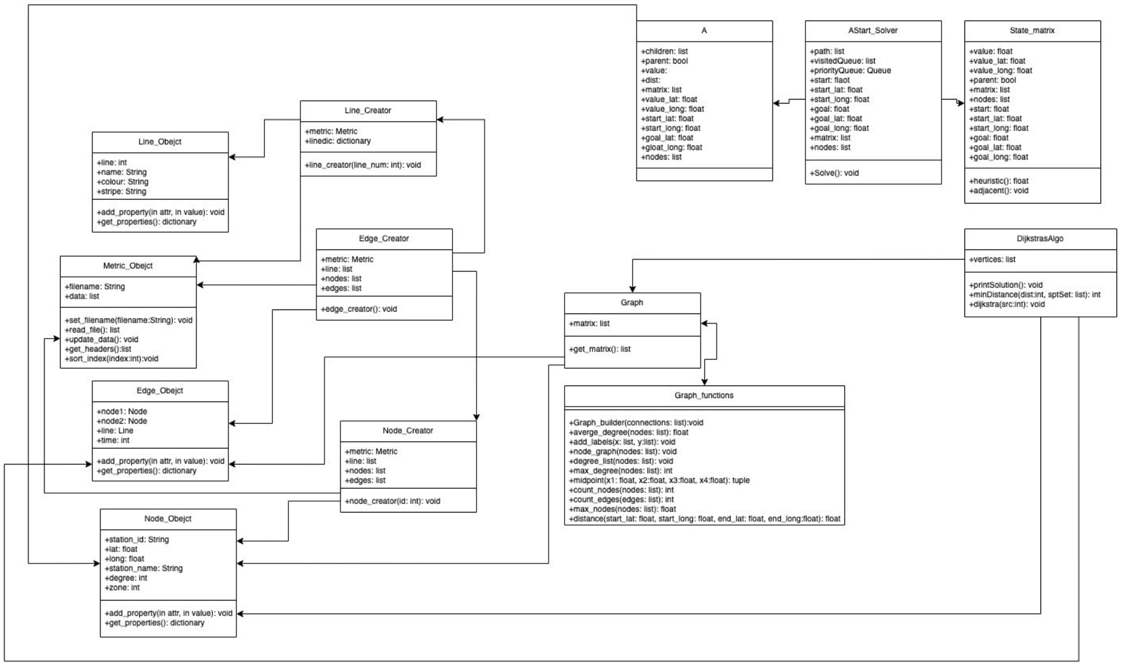
Distribution of Degree vs Nodes



Graph Visualized



Class Diagram



link: https://drive.google.com/file/d/1I9WqMuhGcwMR-BeDRGoo26KsksA7Ctd_/view?usp=sharing

Design Justification

Introduction We created a SOLID adhering design that separates the code for metric extraction and graph creation into two individual layers. Both layers are independent of the other one. The code is self is designed to be readable. Readability was the key feature in mind when designing this project. Each file at the top has input and output statements or a brief description of what the class/set of functions do.

Objects File The Object file contains the python objects that are the core components of the graph builder. Each object is created with the necessary information (ie. Node object has station_id, lat, long, station_name, degree, zone). There is also the feature to add a property to a particular object. (ie. introduce a Busy property to a particular node by calling the add_property function). The metric object file contains some basic functions to read, write, update, and sort CSV files.

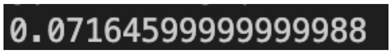
GraphCreation Within the GraphCreation folder, python files to create the graph exist. There are Node, Line, and Edge creators that mass produce Nodes, Edges, and Lines for convenience from CSV file data. By splitting up the 3 files into 3 different objects, we are able to separate the responsibility of each object. For example, NodeCreator is responsible for working with the london.stations.csv file via the Metric Object. This reduces the dependability of the objects on multiple files and makes the code easier to work with.

Algorithms Each algorithm is split into separate files under the same folder called Algorithms. Each algorithm takes in a graph and acts upon that graph, returning the appropriate data. Each algorithm is wrapped in a Class which can be further wrapped up allowing for more features to be added when needed. Please note that these algorithms work under the assumption that the given graph is complete (ie. that there are no islands). My partner and I were informed by our TA that it is safe to assume that the graph is complete.

Benchmark Analysis According to the tests that we ran over multiple sets of data, we conclude that A* is a faster algorithm. Despite having a limited sample size, we can see that A* took 0.2 seconds less than Dijkstra's Algorithm. This is likely due to the heuristic in A* that shows this trend.

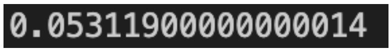
With smaller groups of data, the time taken was too fast for the system to process, giving a result of zero seconds for both algorithms (15 nodes, 14 edges).

With medium group data (100 nodes or so), the time showed a slight difference, but the best result was with the whole sample size, that is being presented here.



```
0.07164599999999988
```

Dijkstra Time



```
0.05311900000000014
```

A star time The executions of these algorithms can also be tested using the Astar_benchmark and Dijkstra_benchmark

WEEK 2

Class Diagram

Self Reflection Questions MacID: chinjerh

1. **Have you done a similar kind of work in the past?** No, I would say that it is the first time that I have worked on a project on this scale and in a team. I have only worked on smaller-scale projects in the previous year and most of my projects have been individual. This project was quite a new experience for me as I had only learned the theory about graphs but never implemented them in a project. In previous years, no course had a project of this scale. Engineering 1P13 had two programming projects with simple algorithms and SFWRENG 2AA4 assignments consisted of only a few classes working together.

2. **How do you feel about this piece of work? What parts do you particularly like? Dislike? Why?** I have very mixed feelings about this piece of work. This project gave me a lot of stress with a lot of debugging but it gave me a lot of satisfaction as well when a feature or an algorithm worked. I particularly liked working with a partner as we came up with ideas together and debated the pros and cons and then executed our plans. I also liked the context of the lab, it was interesting working with a real set of data and using it to find meaningful information. I liked that we got to use the SOLID principles in a project and I was extremely happy when I saw the visualization of all the stations and the connections between them. I disliked the fact that this lab was extremely long. In fairness, although we had three weeks to complete the lab, the implementation of algorithms and integration of different features was extremely tedious and time consuming. Along with this, any logical bug or runtime error took a ton of time to fix.

3. **What is the one thing you particularly want people to notice about your work?** The one thing that I particularly want people to notice is the A star and the Dijkstra Algorithms and the SOLID design principles used in the project. Those two algorithms took quite a lot of time with a lot of debugging to implement. It was easy to find resources online that had these algorithms implemented but their graph structures were completely different and required a lot of modification for them to work in the context of this lab. We had to come up with creative ideas to use the time between stations, latitude, longitude and the station id's themselves to construct the graph and implement the algorithms

4. **What would you change if you had the chance to do this project over again?** If I were to do this project all over again, I would manage my time more efficiently for this project. This project was very time-consuming and took way longer than I initially expected. The majority of the algorithms were new to me and I had to research them before the implementations.

MacID:patek74

1. **Have you done a similar kind of work in the past?** I have not done a similar kind of work in the past. This was the first time I have created a library in python, used OOP in python, and created work from scratch. This was also the first time I used pipenv or any other technology. I was familiar with python syntax and common

types (dictionaries, list methods…etc.). The closed I have come to this kind of work was in my SE 2AA4 class where we designed systems with design patterns while adhering to SOLID principles. However, this was a 3-week long project, while the other was 2 assignments.

2. **How do you feel about this piece of work? What parts do you particularly like? Dislike? Why?** The project itself was very interesting. I enjoyed the parts about reading and creating your own library from scratch. This gave me more insight into how much goes into making and maintaining a library (for all languages). When going about implementation, it was a great deal of work that I struggled to complete. My initial excitement was squashed when the work piled up and I felt like I was not learning, but rather trying to get over it. The parts I particularly liked were the Object creation. It allowed me to create almost a black-box effect where “this thing exists and I use it like this”. What I disliked was the great deal of work and how I felt as if I didn’t have any support (aside from my teammate who struggled with me).

3. **What is the one thing you particularly want people to notice about your work?** I want them to notice our intentions and the results of our work. The intentions were to make an organized and clear piece of work that is modularized. Each folder severs a specific purpose and together, they provide a method to work with csv file data and produce beautiful graphs. The results of our work followed a similar path, albeit a bit different as we rushed to finish the work.

4. **What would you change if you had the chance to do this project over again?** I would change our approach to this project if I were to start over again. I would worry less about “is this right?” and take more action on what I believe is correct. If presented with the same workload, I would spend more time planning so that fewer changes have to be made. I would also keep better communication with my partner so that we know what each of our respective codes is doing. In addition, I would research more into the work I am doing and how I can go about getting there. For example, I would research more into OOP in python and how I can implement the SOLID principles in my design better.

Distribution of Work Krish Patel Objects Folder: Node, Line, Edge, Metric Files GraphCreation Folder: Edge, Line, Node Creators Files Algorithms Folder: Dijkstra's Algorithm UML Diagram file: test_Dijkstras.py Justification of Design Choices

Harsh Chinjer Objects Folder: Node, Edge Graph Creation: Graph, Graph functions Algorithms: A star, Subway Patrol, Urban Planning test cases A star benchmark