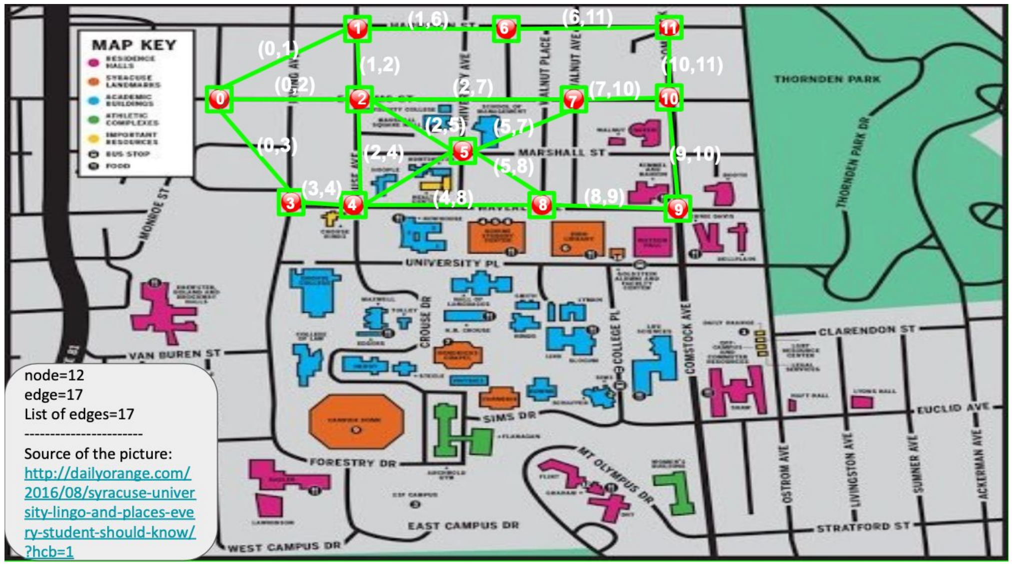
## Executive Summary

### Our approach

With the choice of using Excel or Python as our means of solving the problem our project uses Python. As a result, of this choice, there will be fewer lines of data to work with and its automation skills are needed for this project because the code uses plenty of repetition and creates a graph, which can be easily coded in Matplotlib.

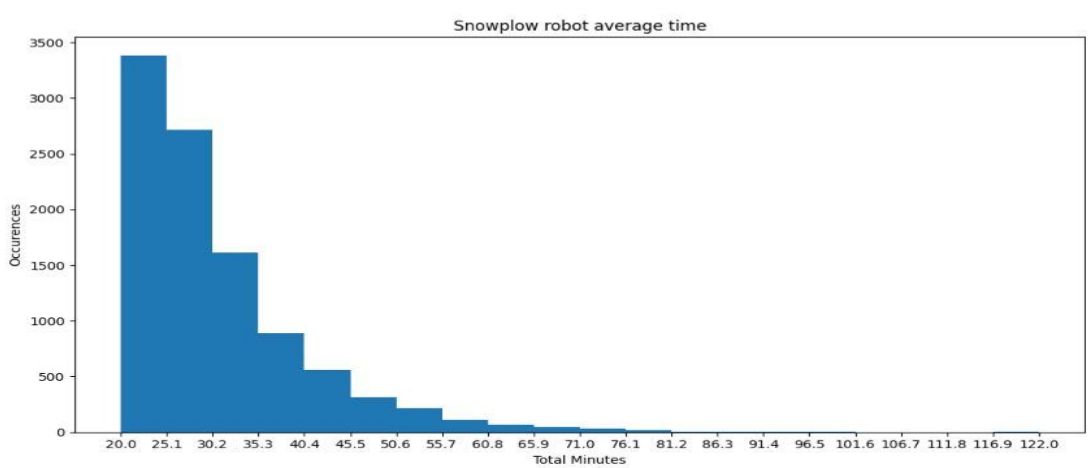
### The Algorithm

The robot, as known, does not have the logic to find the most efficient path, it either randomly selects a snowy road that is connected to its current node or if there are none, randomly selects a cleared road. Therefore, we chose a network on the Syracuse map that included long roads, short roads, and even dead ends, to give our snow plow robot a real test. The Syracuse network provided below consists of nodes that represent the roads and edges that need to be cleared. Therefore, the input to our program was a list of edges representing all the edges in the network. Our snowplow program randomly selected a node to start with and iterated through the logic provided in the problem description until there were no more roads covered in snow. We kept track of the total minutes by incrementing a variable by one every time the robot traveled down a road and also kept track of the backtracks which is when the robot went down a road that was already cleared. Then we simulated the program for 10,000 trials to uncover insight on the efficiency of the snow plow robot.

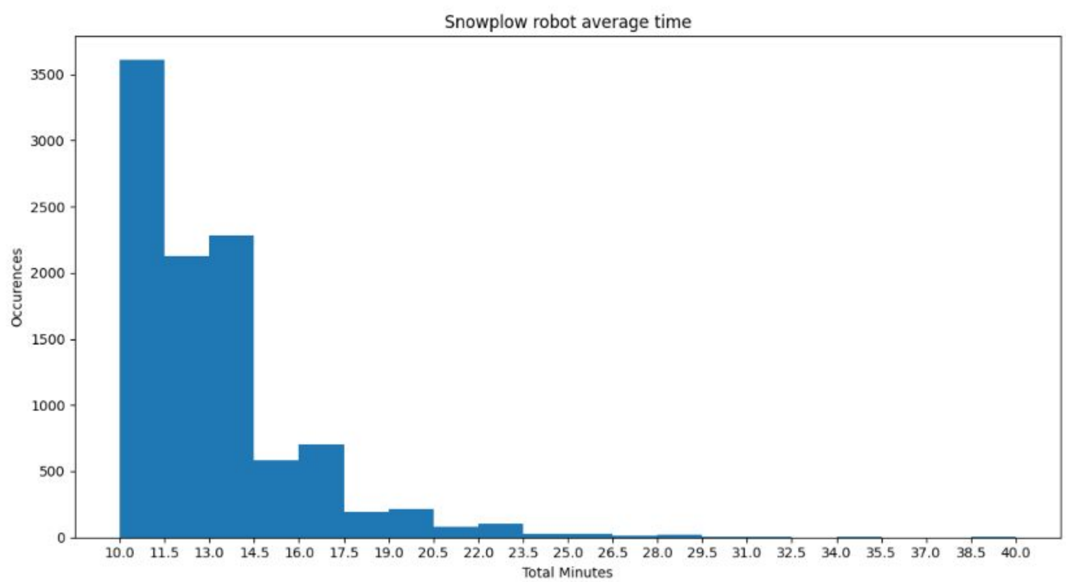



This picture represents the nodes and edges of Syracuse University's map (See Fig 1.)

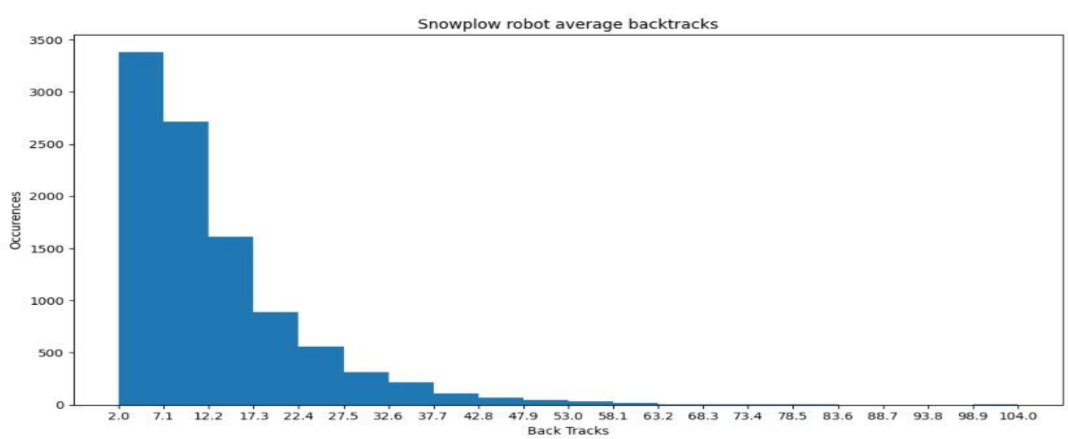
To have a visual representation of our snowplow program we ran a simulation of our robot program on the sample network provided in the problem description and the Syracuse network which was notably larger. The first two histograms show the average time of the robot and the second two show the average number of backtracks.



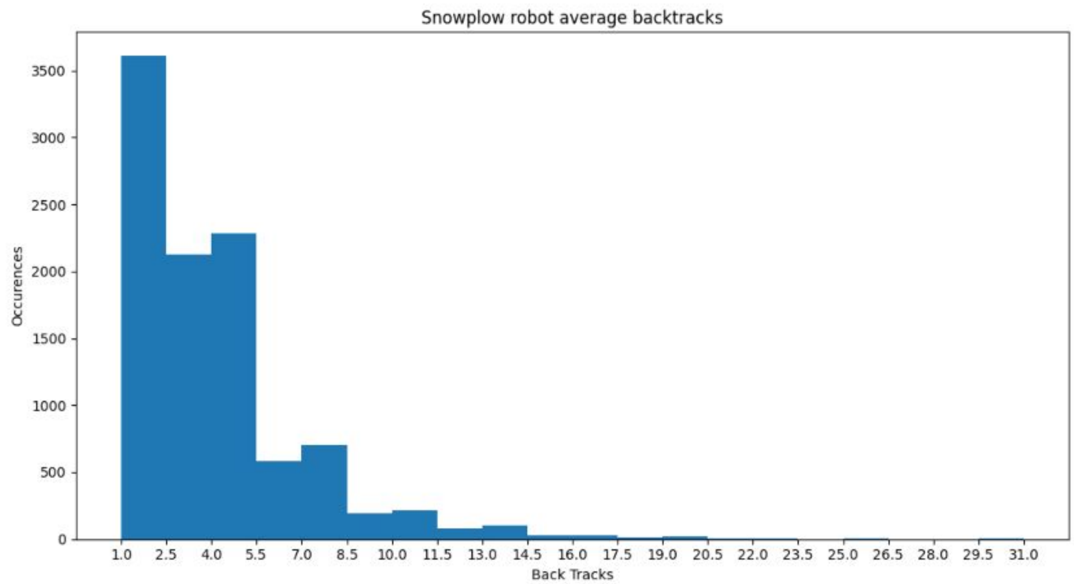
A graph of the average time the Snowplow robot cleared the roads in Syracuse University (See Fig. 2.)




A graph of the average time the Snowplow robot cleared the roads in the sample network (See Fig. 3.)




A graph of the average amount of times the Snowplow robot backtracked cleared roads in Syracuse University (See Fig. 4.)




A graph of the average amount of times the Snowplow robot backtracked cleared roads in the sample network (See Fig. 4.)

**Conclusions and Insights**

The outcome of the histograms were surprisingly more efficient than expected. While looking at the average time for the sample network and Syracuse network, the left skew of the histogram portrays that the snowplow was more likely to clear the network in a faster amount of time. Also, we noticed that the backtrack graphs align with the average time graphs in shape, showing the correlation between the backtracks and average time. Looking more closely at the number of backtracks, we can see that for the sample network, the most backtracks were between 1-2.5 and the most backtracks for the Syracuse network were between 2-7. For future consideration, we would try to implement some robot intelligence in our code to minimize the number of backtracks but with the logic provided in the problem description alone, the robot is somewhat efficient even on large networks such as the Syracuse one.