We intend to obtain the necessary map data from OpenStreetMap (OSM). Henceforth, we seeks to assess the trustworthiness of OSM data. In this regard, we consulted *OSM Science - The Academic Study of the OpenStreetMap Project, Data, Contributors, Community, and Applications*, which provides an academic perspective on the examination of the reliability of OSM data. This publication focuses on the application of OSM data to specific research areas and highlights the impressive scale of the project, with almost 7.5 billion data nodes contributed by 1.8 million users as of March 2022. The authors assert that OSM represents "the most accomplished example of a crowdsourced geoinformation project and of the concept of volunteered geographic information"

Moreover, the publication cites numerous academic studies that have employed OSM data to achieve various research objectives. When scrutinizing the quality of OSM data, scholars have compared it to data produced by world authorities and have found that OSM provides highly accurate statistics. In light of these findings, we are confident in the reliability of OSM data for our project.

Our project endeavors to ascertain the shortest path connecting two points within the university. Analogously, the self-driving automobile proposed in *Application of Dijkstra algorithm in finding the shortest path* shares the same objective but with the added dimension of determining the shortest distance between two regions. The authors have directed their focus towards the utilization of Dijkstra's algorithm and posited that "Dijkstra algorithm is faster than other algorithms for it can calculate the shortest length to every point". The algorithm employs a pyramid tree structure to pinpoint all vertex nodes that fall within the range of interest, as well as a heap to preserve distances and extract nodes with the least distance. These salient features impel us to prioritize testing and deploying Dijkstra's algorithm in our research project.

The Bellman-Ford algorithm is also widely utilized for solving the problem of identifying the shortest path between two points. In *Comparative Analysis between Dijkstra and Bellman-Ford Algorithms in Shortest Path Optimization,* a comparative analysis was performed between the Dijkstra and Bellman-Ford algorithms for optimizing the shortest path. The author of this analysis notes that both algorithms are highly effective for determining the single-source shortest path. Bellman-Ford, as a dynamic algorithm, can efficiently compute the shortest path even in the presence of negative edge weights. Additionally, it performs better on smaller graphs. On the other hand, Dijkstra's algorithm is better suited for larger graphs and positive edge weights, with a time complexity of $O(|E|+|V|\log|V|)$, while Bellman-Ford's algorithm has a complexity of $O(|V|\cdot|E|)$.

Generally, Dijkstra's algorithm is more suitable for real-time applications than Bellman-Ford. In our project, Dijkstra's algorithm will be considered as the preferred option. Nonetheless, we will also conduct experiments to evaluate the efficacy of the Bellman-Ford algorithm, which is still widely recognized as a highly effective algorithm for solving shortest-path problems.

Based on several studies, we found that the Dijkstra algorithm is generally accepted as the most efficient method for optimally solving the 'one-to-one' shortest path problem. However, the authors of *Finding Shortest Paths on Real Road Networks: The Case for A** have performed an empirical study on road network data in California, which has yielded results indicating that “on real road networks, A* outperforms the best implementations of the Dijkstra algorithm by a significant margin.” The superior performance of A* was achieved through the use of spatial coordinates to refine the search for the shortest path. The authors note that the potential of the A* algorithm can be further enhanced by improving its heuristic function. By generating more accurate estimated completion costs, the number of visited nodes and algorithm runtime can be reduced. This observation highlights the flexibility of the A* algorithm and motivates us to improve the A* algorithm for achieving better performance in this project.

Greedy, A-Star, and Dijkstra’s Algorithms in Finding Shortest Path presents a further comparative analysis of the efficiency of Greedy, A*, and Dijkstra, for determining the shortest path in a given graph. The Greedy algorithm, while speedy, may not necessarily guarantee a solution. On the other hand, the A-Star algorithm is relatively more efficient, but its performance is contingent upon complex data. In contrast, the Dijkstra algorithm invariably produces the optimal outcome, making it the ideal choice for shortest path determination. However, it may take longer to reach a solution than the other two algorithms. Despite this drawback, the author unequivocally recommends employing the Dijkstra algorithm for solving problems involving complex searches for determining the shortest path.

In an academic manner, we contend that Greedy is an inadequate algorithmic choice due to its incapacity to yield a solution. Based on the comparative analysis presented in the extant literature, it is our contention that A* and Dijkstra algorithms are the two most highly regarded algorithms for implementation in this project. Our team is eagerly anticipating the practical application of both algorithms in our project to evaluate their effectiveness.

Reference:

1. https://www.mdpi.com/2220-9964/11/4/230
2. https://iopscience-iop-org.ezp3.lib.umn.edu/article/10.1088/1742-6596/2181/1/012005
3. https://iopscience.iop.org/article/10.1088/1757-899X/917/1/012077
4. https://www-tandfonline-com.ezp3.lib.umn.edu/doi/full/10.1080/13658810801949850
5. https://www.semanticscholar.org/paper/Greedy%2C-A-Star%2C-and-Dijkstra%E2%80%99s-Algorithms-in-Path-Wayahdi-Ginting/ddddd926881de4d44a4cfec533057c07b41fe559#citing-papers