This project is almost completely open ended and we encourage you to take on a problem you find interesting. Here are just a couple of examples, but please don't feel limited to these:

1. Although side-streets often provide the shortest path based on distance, we rarely use them because speed limits are usually lower than the speed limit on major roads. In this extension, you can either try to pull in speed limit data from another source OR you could just use "roadType" to make assumptions about speed limits. Then modify your Dijkstra and A* algorithms to use a trip duration rather than distance in finding shortest path. You could go even further and then make predictions about traffic based on road type and time of day (which might again make side-streets a good choice over arterials.)
2. We've talked through a number of algorithms for the Travelling Salesperson Problem. Create a new method which takes a set of vertices and returns a route (not necessarily the best) which visits each vertex and returns back to the start. If you want to go a bit further, you can try multiple solutions we've discussed (e.g., Greedy vs Greedy with 2-opt) and compare the length of path returned. Note, you'll need to use your A* or Dijstra Algorithm to determine distances between vertices. Also, recognize that in real world map data, the path from one vertex to another might include stopping at another vertex in the set. You could manage that by ignoring the multiple visits to the node or by adding a way to mark that node as visited (if you pick that path).
3. When multiple people search Google Maps, it's likely to have some commonality in the searches. For example, if there's a concert in downtown San Diego on Friday night, there will be a number of people searching for directions to the concert venue on Friday afternoon. Each person probably has a unique starting address, but there's likely commonalities in the paths (e.g., multiple paths to get to the main freeway to downtown, followed by the same path from the freeway to downtown). Rather than redoing the entire search every time, you could store previously found shortest paths between vertices. As a result, the first search downtown may take a while, but when the next person searches, they'll use part of the previous solution in your solution.

4. An extension of your choice!

Ultimately, you'll decide what "counts" as an extension and we'll just ask you to report on what you accomplished. We hope you have fun and pick something which is interesting to you and you'd feel proud showcasing to others!