Douglas Cerrato
Final Project Report
Pizza Runners

1. Story/Problem

For our project, we are contracted by a Pizza Company to make a GPS application that will take in multiple addresses with varying distances and connections between each other, and it is our task to have an ending point we need to reach out, and find the fastest route to get to that house following the pathing of the multiple nodes and their distances between them

2. Mathematical Modeling/ Mathematical Abstraction

The mathematical modeling of this code is that it is finding the shortest path between two points using a brute force search, with a few implementations to stop it from going down other routes or paths that are taking longer then the current fastest route we have set. It recursively calls itself to do the brute force search and try every path possible until the fastest route is found. The mathematical abstraction of this code is that it is using a matrix to represent the distances between the different points and using a recursive function to traverse the graph and find the shortest path.

3. Environmental Simulation/Solution Evaluation

A simulation of our project is that we are currently having a test driver try out our application on a day of the route. Below we will have an image attached of the houses we have (listed A-E) and our Pizzeria which will be labeled ‘Z’. We will enter into our algorithm our ending point for the first house we will be delivering to which will be A. From here the algorithm will go down every path recursively and look at the node, then all the connections they have. If the first connection the node has isnt the ending point, it will go down it through brute force search and repeat this process recursively. This will stop the nested recursion call when the connection the algorithm is looking at equals our ending point. Then from here it will work its way backwards from the tree of connections we visited through each node, to see if we can find a faster route to the ending point we are trying to reach. We will also attach below a screenshot of the code and the pathing and routes it takes to find the fastest route. Both images will be found at the end of this paper

4. Best Solution

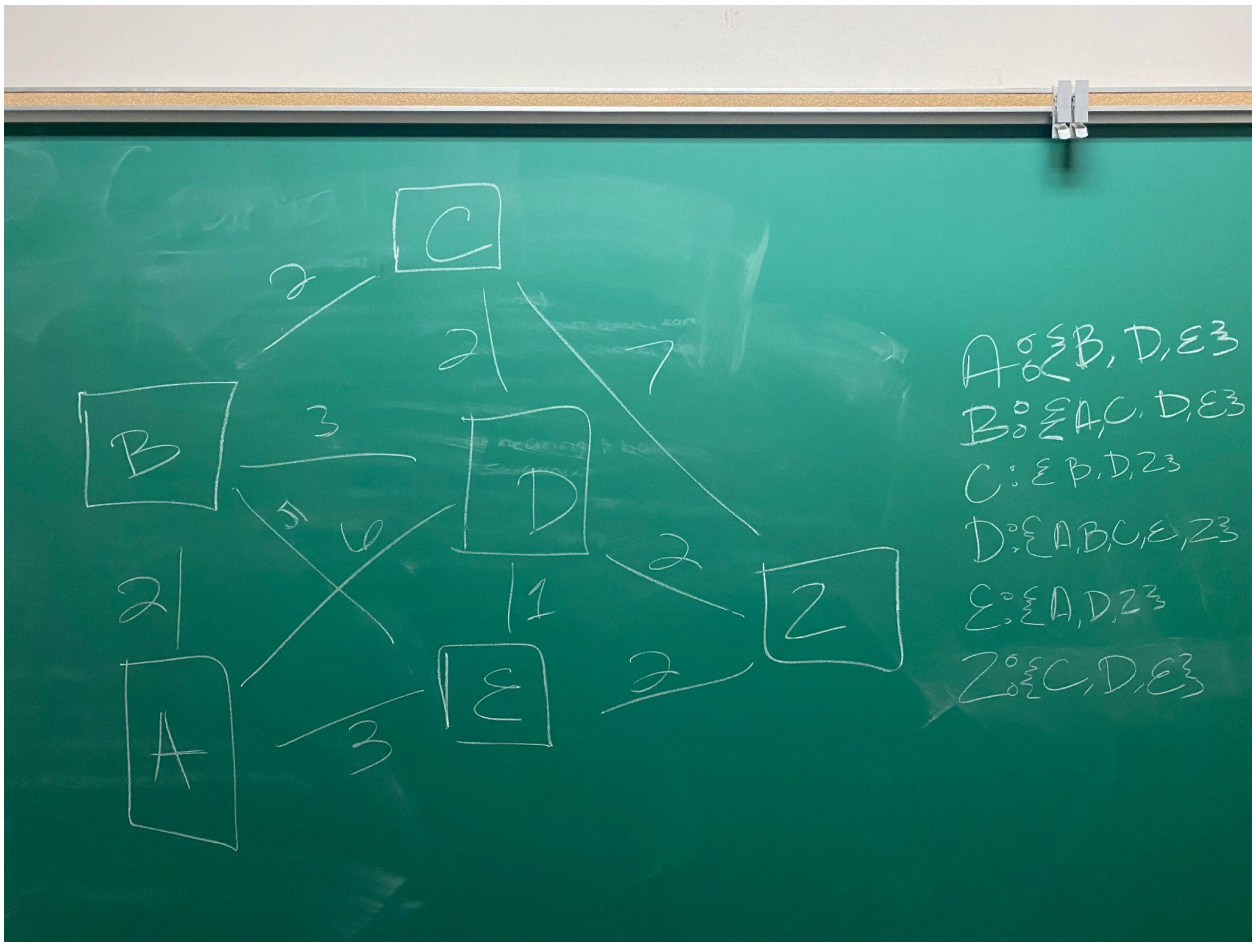
Our best solution for this algorithm would be if the house we are delivering to is the first connection that is connected to the node known as our Pizzeria. This means that the house we are delivering to is the closest house in our radius and we are able to deliver to them the fastest

5. A Better Algorithm

One possible advanced algorithm for traversing a graph recursively is Dijkstra’s algorithm. This algorithm uses a priority queue to keep track of the distances between the starting point and each other point in the graph, and it repeatedly picks the points with the smallest distance from the queue and updates the distances to the other points in the graph. This

algorithm is more efficient than a brute force search because it only considers points that are likely to be part of the shortest path, and it is guaranteed to find the shortest path in a graph with non negative edge weights

This is our reference image for our Simulation. Node Z is our Pizzeria with every other connection being the houses and their relative distances. On the left is our dictionary matrix we used to traverse it.



And this is a screenshot of the path our Algorithm takes to try to find the fastest route to A. It does a brute force search down every path, stopping itself any time it takes longer then our current fastest route