**Team Name:** ByteMe

Connor Thompson

Chandler Garthwaite

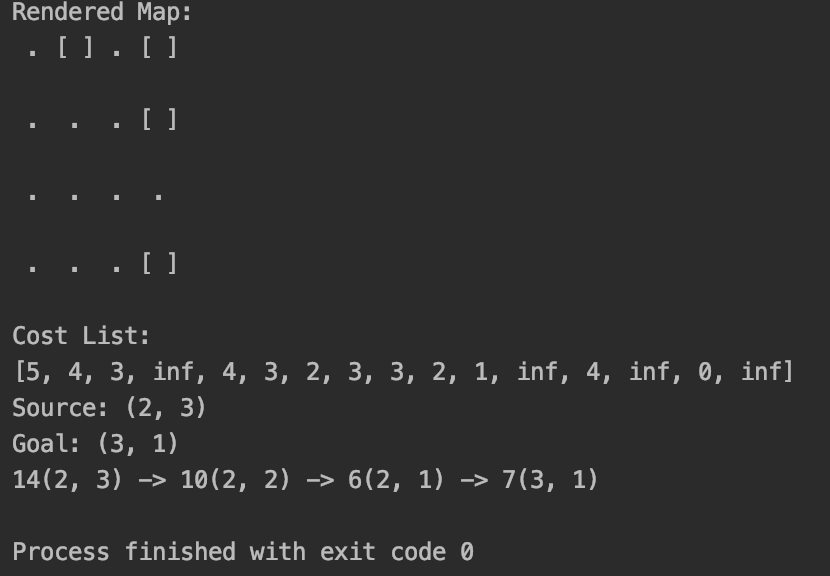
Ian Brobin

Austin Albert

Lab 5 Write Up

1. Our algorithm works by first initializing a few data structures: a list of integers to hold the minimum cost to each node from our source, a priority queue (based on min cost) that determines which unvisited node will be explored next, and a list that will hold the previous node used to reach each node for the shortest path. We then start at the source node and get the cost to reach each of its neighbors, in this instance, our cost is always one since all paths in our graph are weighted equally. For each of the neighbors, we then see if our new cost found is less than the minimum cost to reach that node already stored in our first list data structure. If it is, we update our list with this new minimum cost. In addition, if this new min cost path to the neighbor node is also not an obstacle, we then add it to our exploration priority queue so that we will explore the new node later on in the algorithm, thus also exploring its neighbors. We repeat this process for each node in our graph (cell in our map), and then we are left with a list of the minimum cost to reach each cell in our map from our source as well as a list which we can backtrack through to find a min-cost path from our source to any ending vertex.
2. An example of an admissible heuristic we could use in order to help us implement an informed search of our map would be the Euclidian distance from each cell to the destination cell. This is an admissible heuristic because our robot is restricted to only moving in cardinal directions (N, E, S, W) and not diagonally. Thus, the Euclidian distance is always admissible when used as a heuristic because it will always be less than or equal to the Manhattan distance.

Answers to Question 3 & 4 in Screenshot Below:



1. We spent approximately 7 hours on this lab.