Our maze-solving process involves two main segments – exploration and quick-traversal.

During the exploration phase, the robot employs a depth-first-search of every square in the maze, creating a graph of 256 nodes in which each node represents a square and each edge represents two adjacent nodes without a wall. Using this graph, with each edge weight 1, the robot uses Djikstra’s algorithm to find the shortest path.

The quick-traversal phase will traverse the shortest path between two given nodes, whether it is from start to finish, or from any given node back to the start. We find this path with Djikstra’s algorithm. By maintaining two heaps, one of unvisited nodes and visited nodes. We begin with only the start node in the visited heap, with all other nodes in the unvisited heap, with distances some arbitrarily large number. Then, with the node currently being evaluated, we update its neighbors with distance of current node + 1. We repeat this process, popping the minimum distance node and updating its neighbor’s value to find the shortest path between any two given nodes.