

CS3050 Final Project Report

Algorithm Description - At each step, the robot greedily decides which possible move of the eight spaces surrounding it will bring it closest to its goal without hitting obstacles. The optimal move is determined by performing the pythagorean theorem. The distance between each available space and the finish is the hypotenuse, and the available space with the smallest possible hypotenuse is chosen. This step is repeated until the robot has reached the finish space. In order to prevent the robot from getting stuck in loops, the location two movements ago is checked and preferentially avoided (if another, equally valued move is possible).

Complexity - Each greedy step performs at most 8 calculations, all in constant $O(1)$ time. The max number of possible greedy steps that can be performed is slightly below $O(2n)$ time, therefore the overall complexity of the algorithm is $O(n)$.

How to Run - An executable jar file is included that will open the GUI if the latest version of Java is installed. The source code is included as well, so it could be compiled and run manually. Once the program has opened, click File->Open File, and you will be prompted to select a file. If the file loads successfully, click start, and from then on you will click next step to watch the execution of the program. A key is provided in the application to help understand the grid. Any errors will be displayed in the GUI.

Output - There are two ways to view the results of the algorithm:

1. Watch the robot move at each step on the grid that is provided
2. View the robot's coordinate locations to the right of the grid

Group Contributions - Luke and Jeremy worked together to create the UI and parsing of the input file. The group decided together on the mechanics and pseudo-code of the algorithm that would be used. Jeff implemented the algorithm into the actual code. The whole group worked together to debug and finalize all aspects of the program.

Extra Credit - We chose to implement a GUI for an attempt at extra credit points.