Summary of Project

The program calculates the lowest-cost path from start to goal for an automated car on a given map. It does this by use of either the Iterative Depth-First Search algorithm or the A* algorithm. The first line in the input file is either “IDS” or “A*”, telling the program which algorithm to run.

The rest of the input file contains the map. The map is an nxn grid with the start cell (S) in the top left corner (0, 0) and the goal cell (G) in the bottom right corner (n-1, n-1). All the rest of the cells are either paved roads (R), unpaved roads (D), hills (H), or water (W). I have included three example input files to run the code on. The program always runs on a file located in the current directory called “input.txt”, so to change the input file you must change the names of the files in the directory appropriately.

The car, at any given cell, can move into one of the 8 adjacent cells (including diagonals) according to the following rules:

1. The car may not exit the bounds of the map
2. The car may not enter a water cell
3. The car also may not diagonally enter a cell if a water cell is adjacent both to that cell and to the car’s current position. For example, if there is a water cell to the right of the car, the car may not travel diagonally right-up or right-down (and it obviously can’t travel right either, by rule 2).

In addition, each movement incurs a cost. Moving into a paved cell incurs a cost of 1, an unpaved cell a cost of 3, and a hill a cost of 10. The program finds the lowest-cost path from the start cell to the goal cell.

How to Run

On a Linux system, download the files so that they’re all in one directory. Open a terminal and `cd` to that directory. Type `make` and hit enter. Then type `./a.out` and hit enter.

That’s it! You should now find a new file in the directory called “output.txt”. The file contains the path that the car should take, with each move separated by a hyphen. R=right, L=left, D=down, U=up, and diagonal moves are denoted with the two letters of the moves making up the diagonal. So for example, a right-up diagonal move is denoted RU.

You will also have a new executable file called “a.out” in the directory, along with many object files. If you wish to run the program again without recompiling, leave these files in the directory and just enter `./a.out` again in the terminal. If you wish to get rid of these generated files, enter `make clean`.