CMPT 310 Assignment 1
Documentation

Search procedure

In this assignment, I use uniform-cost search to find the least cost path on the grid, given a start location and a goal location. Uniform-cost search is an uninformed search which does not require complete information of the whole problem. The problem defined by the assignment starts with a $n \times n$ grid fulfilled with entries of cost. And we only have the current state(i.e., the cost of the current entry) and the successors' states(i.e., the costs of the neighbors of the current entry). Since we do not have the prior information about any other entry besides the neighbors of the current node, the heuristic function that calculates the approximated cost of the problem can not be established. Therefore, uninformed search is not suitable for this problem, and we use informed search.

The program is initialized with information of the start location(i.e., the cost and the coordinations on the grid), then builds the search tree by setting the start location as the root. Uniform-cost search expands the node with the least accumulated cost from the start location and marks the expanded node as being explored. The program recursively expands the node with the least cost among all the frontiers of the search tree until the goal location appears in the frontiers of the tree, and it has the least cost among the others.

Since the algorithm expands the nodes in increasing order of $f(n)$ where $f(n)$ is the cost to get to the node n from the start location, uniform-cost search is optimal in finding the least cost path, that is, the path found by the search must has the least cost among all other possible paths that reach to the goal location. Uniform-cost search is complete if the step cost exceeds or equal to some positive constant ε. If the grid has some entries(nodes) with the cost of 0, then the algorithm will gets stuck in an infinite loop, in which the node with zero cost forces the path searcher to choose itself to expand, but does not contribute to the accumulated cost for the path. Therefore, the path searcher will keep traversing between the node with zero cost and one of its neighbors, back and forth.

High level description of the program

The program consists of two parts: $main()$ and $path-find()$. Given the grid size, start location, goal location, and values of grid, $main()$ calls $path-find()$ to output the least cost path. The major component of $path-find()$ is a while loop in which the algorithm searches the neighbors of the current node. During each iteration, the program appends the location of each neighbor to the paths, and adds the cost to get to each neighbor upon the total cost for the current path. The cost and the path are stored in a priority queue, in the format of $(cost, [path])$. When the expansion of the current node reaches to the goal location, the path will be returned to the $main()$ for printing out.

Notes about the program

In the problem defined by this assignment, the cost to get to the goal location from any of its neighbors is the same(i.e., the cost of the goal location itself). However, in real-life problems such as choosing an optimal path for travelling from one place to another, the cost to get to the destination from its adjacent cities may be different. The nice thing about the program is that it generalizes the problem-solving for finding the least cost path, and applies to the both situations mentioned above.