**Hill Climbing Search**

Pseudo Code:

HillClimb(M,S,E) // M is map, S is start node, E is end node

Generate H values()

Set Current Node to S

while(Current Node is not null)

set Current Node to visited

if(Current Node is E)

Create Path()

return

else

set Next Node to null

foreach(neighbor node in Current Node)

if(neighbor node's h < current lowest h of neighbors)

set Next Node to neighbor node

set Current Node to Next Node

Properties:

* Complexity:
  + Time Complexity: O(∞) since it is not complete
  + Space Complexity: O(n) where n is number of local edges
* Completeness: It is not guaranteed to find a goal state if one exists.
* Admissibility: If it finds a solution the solution may not be the best solution
* Irrevocability: Yes

Heuristic:

The heuristic that was used for this search is called the Manhattan Distance. It is simply the x distance from the current node to the goal node plus the y distance from the current node to the goal node. This was choose for this search because we didn’t move in any diagonal directions, therefore, we wanted the minimum cost to move from a node to the end node. The heuristic function and evaluation function are below.

H(n) = |current node x value – goal node x value| + |current node y value – goal node y value|

F(n) = H(n)