

Question 4

Answer 1

A heuristic of my design, the one that I have chosen is one that takes the MIN of dx, or dy.

where $dx = \text{currentx} - \text{endx}$, and $dy = \text{currenty} - \text{endy}$.

The reason this heuristic is admissible because it never estimates the path to the destination, it tries to choose always the lesser of the dx or dy values and hence will never overshoot the destination.

Question 2

Suppose you were instead given several end locations, and your task was to place a wire that began at the start location, and ended at any one of the end locations the algorithm prefers. Note that all, none, or a subset of the end locations may be reachable. One obvious extension of the Manhattan-distance heuristic would be the Manhattan distance from the new wire's current ending location to the closest end location:

$$h((x, y)) = \min |x - x'| + |y - y'|$$

Answer 2

part (a)

$$H(x,y) = \text{for all } (x,y) \quad \min(|x - x'| + |y - y'|)$$

No this heuristic is not admissible,

part (b)

It's not admissible as at every node it will try to find the minimum Manhattan distance to all end locations.

The smallest Manhattan distance may not be best heuristic as it may lead to a path which leads to a blocked path.

part (c)

We can improve this heuristic by adding `blocked_count` to it, where blocked count is the count of the blocked tiles on the rectangle formed by the Manhattan distance from the current location to the destination. So the new heuristic would become

$$H(x,y) = \text{for all } (x,y) \quad \min(|x - x'| + |y - y'| + b')$$

b' = number of blocked tiles in the rectangular area formed by the Manhattan distance.

This heuristic will give a new heuristic which provides the best Manhattan distance that increases the probability of reaching the destination. As it always chooses a region that has fewer number of blocked tiles and hence increasing its chances of reaching the destination.

Answer 3

I built my code in a step by step manner. I added a lot of debug statements which show the state of the program at various points of execution. I have included the `-graph` option to show the current graph. Also I have included the `-D` option that makes it easy to show the debug output. Also I have structured the code in a place which enables and makes changes easier.