```
// CPCS 4610
// Jakob Balkovec
// HW1.Q2

// The heuristic needs to be admissible and consistent.
// - Admissible: should not overestimate the actual cost to the goal.
// - Consistent: must satisfy the triangle inequality, ensuring
//    the estimated cost from the current node to the goal is no greater
//    than the cost of reaching a neighbor plus the estimated cost from the neighbor
 to the goal.

// Pulled from: [https://en.wikipedia.org/wiki/Consistent_heuristic]

// Triangle inequality | Cons: h(n) <= c(n, n') + h(n')
// Admissible: h(n) <= h*(n)

// Idea: Use straight line distances or some logical esitmates to guide A*
//       effectively towards the goal while misleading DFS and UCS.

// Simple Pseudo Code:

function heuristic (current_node, goal_node):
  // Input:
  //  current_node: The current node in the graph.
  //  goal_node: The goal node in the graph.

  // Output:
  //  (int) The estimated cost to reach 'goal' from 'current'.

  // Straight line distance between two points
  // Euclidean distance, could use others like Manhattan or Chebyshev
  // Pulled from the powerpoint slides (Wednesday)
  return sqrt((current_node.x - goal_node.x)^2 + (current_node.y - goal_node.y)^2)

// Mock results:
// Abbreviated 'heuristic' to 'h' for brevity
// Nn = current_node, N10 = Goal state
//
// h(N1, N10) = 10
// h(N2, N10) = 8
// h(N3, N10) = 7
// h(N4, N10) = 6
// h(N5, N10) = 5
// h(N6, N10) = 4
// h(N7, N10) = 3
// h(N8, N10) = 2
// h(N9, N10) = 1
// h(N10, N10) = 1
```