Homework 17

Due 11/09/16

November 3, 2016

The *rook-path problem* accepts an array of n 2D points *data* and returns the length of the shortest *rook-path* from $data[1]$ to $data[n]$, where a *rook-path* is a sequence of moves that start on a point in *data* and move to another point in *data* that is horizontal or vertical.

For example, if $data = \{(0,\ 0), (10,\ 0), (10,\ 1), (0,\ 2), (1,\ 2), (1,\ 1)\}$, the shortest rook-path from (0, 0) to (1, 1) would have length 4: (0, 0) – (0, 2) – (1, 2) – (1, 1). There is another rook-path of length 20 from (0, 0) to (1, 1), but length 4 is the shortest rook-path. Note that the rook can't move to (0, 1) or (1, 0) from (0, 0), because these points are not in *data*.

Describe an efficient algorithm to compute the shortest rook-path in a given array of points.