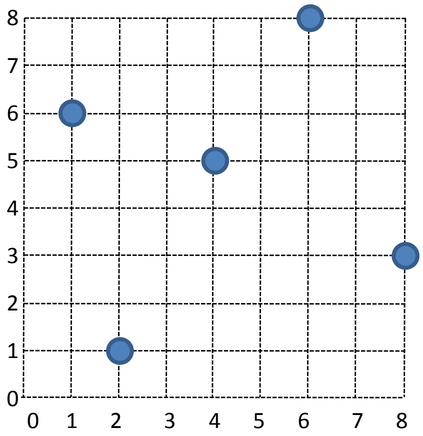
NOTE: Parts A and B are independent from each other.

A. (2 points total) Consider the Rectilinear Steiner Minimum Tree (RSMT) routing problem.

i. (0.25 points) Define the problem, its objective and constraints. What is the Hanan gid?

ii. (0.25 points) What is the time complexity of an algorithm that can find the optimal solution to all instances of the RSMT problem?

iii. (0.25 points) Show the Hanan grid for the following terminals (double check the grid coordinates to make sure you do not make an error when copying the grid).



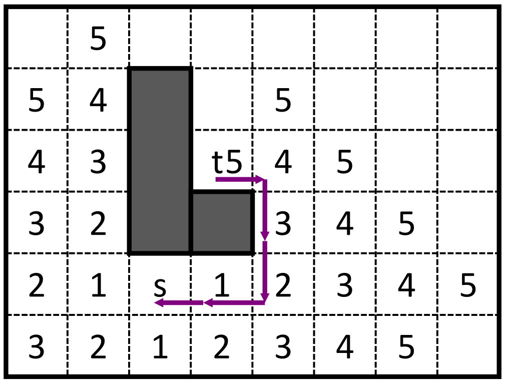
iv. (0.5 points) Give an optimal RSMT solution for the above terminals.

v. (0.25 points) What is the length of the tree in your solution?

vi. (0.25 points) What makes the Hanan grid special? In other words, why is it useful?

vii. (0.25 points) Give an example objective function for which Hanan grid not useful.

B. (2 points total) In an FPGA we have tiles in the architecture that lend themselves naturally to a grid-based routing with regular grids. Vaughn has come up with a rough sketch of an algorithm to find the shortest path from a source to a destination node in a 2-terminal net. He calls his method the "maze routing" or "wave propagation" algorithm. The figure below shows an example of how his algorithm is supposed to run to find a shortest route from node "s" to node "d".



	5						
5	4	▨		5			
4	3	▨	t5	4	5		
3	2	▨	▨	3	4	5	
2	1	s	1	2	3	4	5
3	2	1	2	3	4	5	

The shaded regions are routing blockages. Cells with equal numbers show a "wave front" at some point in the algorithm. The algorithm first generates the wave from "s" and labels cells in the immediate vicinity as "1". Then the wave expands and cells are labeled "2", and so on until we reach "t" with a label (distance) of 5.

i. (0.25 points) If the edges between grids are not weighted, what known algorithm should he use for his method? Write the pseudo-code of the algorithm.

ii. (0.25 points) what is the time complexity of this algorithm?

iii. (0.75 points) If we keep routing all two-terminal nets independently, we end up with grid locations that have more nets than routing resources (over-congested regions). How should he handle this problem? Is the algorithm in part (i) still going to be useful? Explain why.

iv. (0.75 points) How would you modify the wave propagation algorithm to handle multi-terminal nets? (assume that grids are not weighted). A pseudo-code is not needed. Just the general strategy. Note that an algorithm that runs the wave propagation starting from the source node multiple times and simply combines the routes into one solution is not efficient and generates poor solution qualities.