Chris Liow
ECE 368
IPA 5

Report

My Results

Test File	N	Number of Turns
river.in0	3	4
river.in1	6	9
river.in2	8	10
river.in3	15	22

For this IPA, I initially tried creating an adjacency matrix and then running it through Dijkstra's algorithm to find the minimum cost. I ran into too much trouble setting weights and dealing with my memory leaks, so I decided to approach the problem differently since I had 3 extra days to work on the assignment due to the extension. The algorithm used in my code traverses the river based on a set pattern until it gets to the rightmost side of the river. For traversal, I prioritize moving to the right (obviously) with moving to the left being the last option. To find the ideal path, I allow myself to move freely up and down when it does not cost me any rotations. I decide to move to the right by checking if the node one over from my current position is connected with a vertical rod (basically, I want to move gradually to the right while getting as many free movements as I can). Conditional checking for my algorithm involves making sure I do not get caught in infinite loops, aka, traversing in a circle. To prevent this, I modify my traversal algorithm when I detect that I am visiting the same nodes over and over again.

Since I do not construct an adjacency list or matrix, the only memory I allocate is for the array to hold what I read in from my input file. Therefore, my space complexity is O(V). As for time, my complexity is O(V) as well since I run through one iteration to traverse each node or vertex.