Name: Sang Hun Kim
PUID: 0025808690
4/29/2017

Programming Assignment 6

From this assignment, we were asked to find the least amount of turns of boards to cross the river from one side to another. The input file was given with the number of rows of poles (M), the number of columns of poles (N), and the locations of boards available to cross the river. 1s indicate the available boards located on specific location of pole, and 0s indicate that there is no board. The very first thing that I did is to create the multiarray of my own struct, which contains the locations of boards on each pole from the given input. This procedure takes O(N + M) time, because it memory allocates for each row of poles.

Next, I traversed all the nodes and found the amount of turns that is required to cross the river for each row. This procedure takes time complexity of O(N * M) because it has to go through all the poles that are located in specific row and column. Its space complexity is O(1) because this step can be done in single stack frame. It does not require any stack frame since there is no recursion.

After finding all the turns required for all the poles located in each rows and columns, I called the function called update_rotation_count in order to update any number of turns that can be reduced to satisfy the least amount of turns of boards to across the river. This function looks at every direction that a pole can reach except to the East side since it has already been calculated from the second step. The time complexity of this function is O(1) because there is only if, and else if cases that searches all the possible direction that the pole can reach and updates number of rotations within that if and else if cases. However, the space complexity, in worst case, can be done in O(M) because whenever the number of rotations are updated from the specific pole due to another pole that has the board, that pole with the board now needs to be updated for its number of turns, in which there is a recursion that updates the poles' characteristics that are not on the same row. Thus, the space complexity is O(N).

Finally, retrieve_minimum_val function is called to retrieve the least amount of rotations that can be done to cross the river. This function only looks at each row's last pole's number of rotations and get that value. This has the time complexity of O(N) and space complexity of O(1).

Overall, my program has the time complexity of O(N * M), in which it can be represented as $O(N^2)$ and space complexity of O(N) as worst case.

I believe this program could have been improved if I used graphs, instead of just multidimensional array. Typically, if I were to use dijkstra algorithm to figure out the shortest path to reach the other side of river, it would have been much simpler and easier with less time complexity and space complexity.