For my program I chose to implement the longetst_path algorithm iteratively with a recursive and iterative depth first search subroutine. The choice to implement the subroutine recursively was a no-brainer, as every point along the path would have to be selected and checked for paths that could branch off of it. Because of this branching pattern I thought that recursion would work best. The main function works iteratively with two loops to cover every row and column. These two loops being based off of rows m and columns n, multiply to make the spoken of N, so that the big-O complexity is O(N).

The issue I had and still have is with the multiple paths same length requirement, throughout my testing when there are multiple same length paths in the torus, my program returns one of the paths. From my understanding no particular path is supposed to be returned, but hey.

The big-O complexity is O(N) with N being m*n, as is required by the assignment.