Time and Space Complexities: (methods used to solve the problem)

4. Directed Or Undirected Algorithm

- Time Complexity: O(n^2) where n is the numbers of vertices in the matrix. Each of the two nested loops run n times so O(n^2).
- Space Complexity: O(n^2) where n is numbers of vertices in the matrix since the algorithm stores the input matrix that is n*n

5. Paths Of Length 7

- Time Complexity: O(d^7) where d is the average number of neighbors each node points to. This algorithm explores paths of length 7 from the starting node. So at each depth, there are d choices on each depth and 7 depths are explored.
- Space Complexity: O(V+E) where V is the entry per vertex and E is the total edges across all lists. Space is mainly used to store the graph.

6. Circular Graph

- Time Complexity: O(n) where n is number of vertices because we use a loop through the formatted input and build a list. Many of the operations are either O(n) or O(1) which results in O(n) time complexity.
- Space Complexity: O(n) where n is number of vertices because vertex list stores n elements. Graph n nodes and edges which is O(n). The memory stores the lists and graph structure.