1. **Single Source Shortest Path (SSSP) : [Unweighted + BFS]**
   How short a distance from all nodes to a selected node.
   It will be by using BFS because it traverses level wise which is the shortest path. For this we need a level array.
   Level[source] = 0
   While exploring unvisited node we need to update that node's level too.
   Level[adjacent] = Level[prev/head] + 1
   So distance of all nodes from source will be level[source] - level[selected]

   For **Weighted** graph, BFS won't work, for this we need **Dijkstra Algorithm**

2. **SSSP in Code : Implementation**

3. **Grid Traversal : 2D Grid/ Maze**
   First off all Convert the maze into a graph.
   Point out source and destination.
   Without maintaining adjacency list just by checking up, down, righ, left we can traverse.

4. **Grid Traversal in code : Implementation**
   Problem solving from CSES : Labyrinth / Maze