I. Adjacency Lists / Matrices

A) Write code to find an element in a matrix where the elements in each row and column are in a non-decreasing order

Example:

{ 2, 14, 26, 37, 43, 51, }

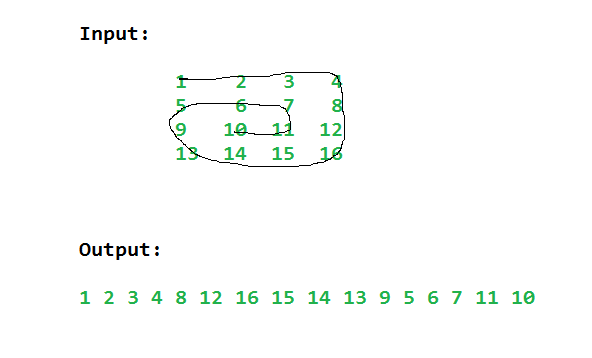
{ 4, 16, 28, 38, 44, 54, }

{ 6, 18, 30, 39, 45, 57, }

{ 8, 20, 32, 40, 46, 60, }

{ 10, 22, 34, 41, 47, 63, }

{ 12, 24, 36, 42, 48, 66, }

B) Spiral Matrix: Write code that traverses and prints out a matrix in a spiral form 

II. Trie

A) Advantages of a Trie

B) Dis-advantages of a Trie

C) Use cases for a Trie

III. B-Tree

A) List Properties of B-Tree

IV. Graph

A) Given a list of edges in a graph or "Forest," write code to find the distinct amount of "trees" or separate nodes (e.g. other metaphors, islands in an ocean, trees in a forest, disconected components)

Input : edges[] = {0, 1}, {0, 2}, {3, 4}

Output : 2

Explanation : There are 2 trees

0 3

/ \ \

1 2 4

V. Dijkstra - Proofs

A) Does this algorithm work for negatives - why or why not?

B) Does the shortest path change when weights of all edges are multiplied by 10?

C) Given a directed weighted graph and the shortest path from vertex ‘s’ to ‘t’ => D(s,y),

If weight of every edge is increased by 10 units, does the shortest path remain same in the modified graph?