Spiral Print Problem Matrix LeetCode

Given a matrix, convert it into a spiral representation by traversing the elements in a spiral pattern. Return the spiral representation as an array.

Initial Matrix

13 → 34 → 33 → 43

↓

39 → 54 → 11 17

↑ ↓ ↓

18 22 ← 27 29

↑ ↓

19 ← 23 ← 88 ← 10

Output: 13 → 34 → 33 → 43 → 17 → 29 → 10 → 88 → 23 → 19 → 18 → 39 → 54 → 11 → 27 → 22

Function to convert a matrix into spiral representation

- The **spiralPrint** function takes a matrix as input and converts it into a spiral representation.
- The function uses four variables **startRow**, **startCol**, **endRow**, and **endCol** to keep track of the boundaries of the current submatrix to be traversed.
- The function uses a **while** loop to iterate while there are still valid rows and columns to traverse.
- Within the loop, it traverses the top row from left to right, the right column from top to bottom, the bottom row from right to left (if applicable), and the left column from bottom to top (if applicable).
- The traversed elements are stored in the **ans** array.
- The function returns the **ans** array.

Time Complexity: The time complexity of the spiralPrint function is O(rows * cols) since it visits each element in the matrix exactly once.

Space Complexity: The space complexity is O(rows * cols) as we store the spiral representation in the ans array.