Here’s the rearranged list of matrix problems, ordered from **linear complexity (simpler)** to **non-linear complexity (more complex)**:

**Linear Complexity Problems (O(m \* n))**

1. **Transpose Matrix** (LeetCode #867):
   * Return the transpose of a matrix.
2. **Set Matrix Zeroes** (LeetCode #73):
   * Modify the matrix to set rows and columns to 0 if an element is 0.
3. **Flood Fill** (LeetCode #733):
   * Perform a flood-fill operation starting from a given pixel.
4. **Spiral Matrix** (LeetCode #54):
   * Traverse a 2D matrix in spiral order.
5. **Diagonal Traverse** (LeetCode #498):
   * Traverse a matrix in diagonal order.

**Logarithmic Complexity Problems (O(log(m \* n)) or Search-Specific)**

1. **Search a 2D Matrix** (LeetCode #74):
   * Search for a target in a sorted 2D matrix.
2. **Search a 2D Matrix II** (LeetCode #240):
   * Search for a target in a matrix with rows and columns sorted.
3. **Kth Smallest Element in a Sorted Matrix** (LeetCode #378):
   * Find the k-th smallest element in a sorted matrix using binary search.

**Moderate Complexity Problems (O(m \* n) with Backtracking or Recursion)**

1. **Word Search** (LeetCode #79):
   * Determine if a word exists in a grid using backtracking.
2. **Rotate Image** (LeetCode #48):
   * Rotate a 2D matrix 90 degrees clockwise in place.
3. **Game of Life** (LeetCode #289):
   * Simulate the Game of Life on a 2D grid.
4. **Matrix Block Sum** (LeetCode #1314):
   * Compute the sum of all elements in k x k blocks.

**Dynamic Programming Problems (O(m \* n))**

1. **Unique Paths** (LeetCode #62):
   * Find the number of unique paths in a grid.
2. **Unique Paths II** (LeetCode #63):
   * Find unique paths in a grid with obstacles.
3. **Minimum Path Sum** (LeetCode #64):
   * Calculate the minimum path sum from the top-left to the bottom-right.
4. **Maximal Square** (LeetCode #221):
   * Find the largest square of 1s in a binary matrix.
5. **Count Square Submatrices with All Ones** (LeetCode #1277):
   * Count all square submatrices containing only 1s.

**Graph and Connectivity Problems (DFS/BFS, O(m \* n))**

1. **Number of Islands** (LeetCode #200):
   * Count the number of islands (connected 1s) in a grid.
2. **Surrounded Regions** (LeetCode #130):
   * Capture all regions surrounded by X on a board.
3. **Path with Maximum Gold** (LeetCode #1219):
   * Find the maximum gold collected by traversing a grid with constraints.

**Advanced Problems (O((m \* n) log(m \* n)) or Higher)**

1. **Longest Increasing Path in a Matrix** (LeetCode #329):
   * Find the longest increasing path in a matrix using DFS with memoization.
2. **Sparse Matrix Multiplication** (LeetCode #311):
   * Multiply two sparse matrices efficiently.
3. **Word Search II** (LeetCode #212):
   * Find all words in a grid using a Trie for prefix searching.

This arrangement moves from problems with simple linear traversal to those involving recursion, dynamic programming, and more complex graph or optimization techniques. Let me know if you'd like detailed solutions for any!