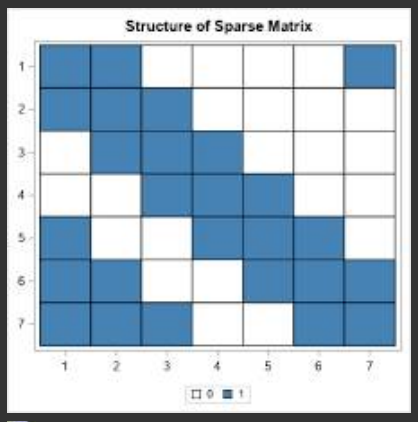
ChatGPT - DSA Mastery - Python

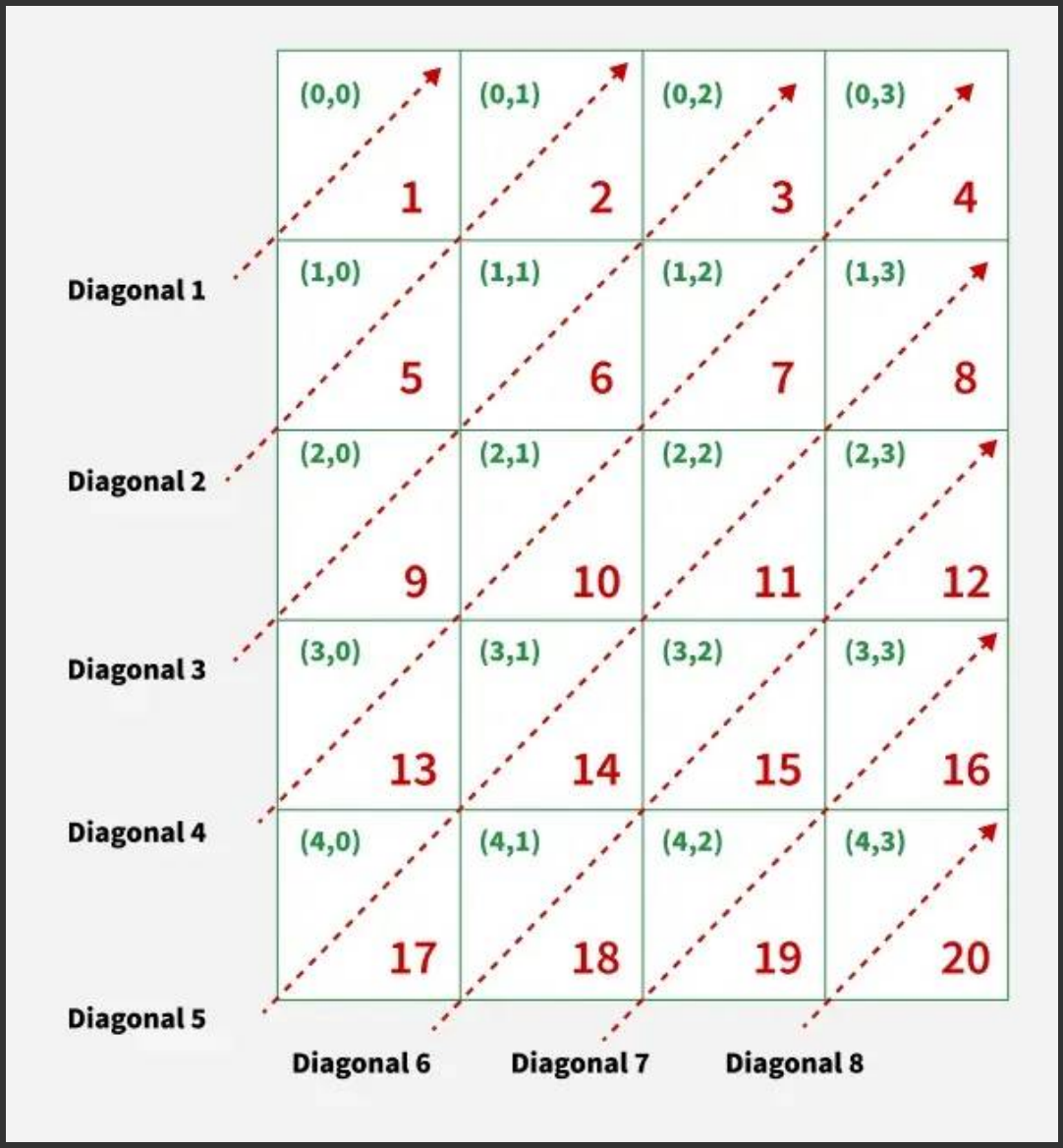
Matrix

Here is your **DSA Mastery – Matrix Cheatsheet (Python)**
Super-clean, pattern-based, interview-centric, with templates used in LeetCode + CP.

✅ Matrix DSA Mastery – Patterns, Tricks, Templates (Python)



https://www.researchgate.net/publication/311458292/figure/fig4/AS%3A862875342106624%401582736921335/The-2D-shape-grid-mapping-algorithm.jpg?utm_source=chatgpt.com



⭐ 1. Matrix Traversal Patterns

1.1 Row-wise & Column-wise Traversal

```python
for r in range(m):
    for c in range(n):
        print(mat[r][c])
```

1.2 Boundary-safe Directions Template

Most used in BFS/DFS/Grid Problems.

```python
DIRS = [(1,0), (-1,0), (0,1), (0,-1)]
for dr, dc in DIRS:
    nr, nc = r + dr, c + dc
```

```
    if 0 <= nr < m and 0 <= nc < n:
        ...
```

⭐ 2. DFS / BFS on Grids (islands, flood fill, components)

DFS Template

```python
def dfs(r, c):
    if not (0 <= r < m and 0 <= c < n) or mat[r][c] != 1:
        return
    mat[r][c] = -1   # visited
    for dr, dc in DIRS:
        dfs(r+dr, c+dc)
```

BFS Template

```python
from collections import deque

q = deque([(sr, sc)])
visited = set([(sr, sc)])

while q:
    r, c = q.popleft()
    for dr, dc in DIRS:
        nr, nc = r+dr, c+dc
        if 0 <= nr < m and 0 <= nc < n and (nr, nc) not in visited:
            visited.add((nr, nc))
            q.append((nr, nc))
```

⭐ 3. Matrix Search Patterns

3.1 Search in Row-Col Sorted Matrix

```python
r, c = 0, n-1
while r < m and c >= 0:
    if mat[r][c] == target:
        return True
    elif mat[r][c] > target:
        c -= 1
    else:
        r += 1
```

3.2 Binary Search Matrix (LC 74)

Flattened index trick:

```python
l, r = 0, m*n - 1
while l <= r:
    mid = (l+r)//2
    row, col = divmod(mid, n)
    if mat[row][col] == target: return True
    if mat[row][col] < target: l = mid+1
    else: r = mid-1
```

⭐ 4. Matrix Rotation & Transformations

4.1 Rotate 90° Clockwise

```python
# transpose
for i in range(n):
    for j in range(i+1, n):
        mat[i][j], mat[j][i] = mat[j][i], mat[i][j]

# reverse each row
for row in mat:
    row.reverse()
```

4.2 Flip Matrix

```python
# horizontal flip
mat.reverse()

# vertical flip
for r in mat:
    r.reverse()
```

⭐ 5. Prefix Sum Matrix (2D CumSum)

Build prefix matrix

```python
ps = [[0]*(n+1) for _ in range(m+1)]

for r in range(m):
    for c in range(n):
        ps[r+1][c+1] = ps[r][c+1] + ps[r+1][c] - ps[r][c] + mat[r][c]
```

Query sum of submatrix (r1,c1) → (r2,c2)

```python
def query(r1, c1, r2, c2):
    return ps[r2+1][c2+1] - ps[r1][c2+1] - ps[r2+1][c1] + ps[r1][c1]
```

⭐ 6. DP on Matrix Patterns

6.1 Minimum Path Sum

```python
dp = [[0]*n for _ in range(m)]
dp[0][0] = mat[0][0]

for r in range(m):
    for c in range(n):
        if r == 0 and c == 0: continue
        dp[r][c] = mat[r][c] + min(
            dp[r-1][c] if r > 0 else float('inf'),
            dp[r][c-1] if c > 0 else float('inf')
        )
```

6.2 Maximal Square

Maintain dp of side length:

```python
dp = [[0]*(n+1) for _ in range(m+1)]
max_side = 0

for r in range(1, m+1):
    for c in range(1, n+1):
        if mat[r-1][c-1] == "1":
            dp[r][c] = 1 + min(dp[r-1][c], dp[r][c-1], dp[r-1][c-1])
            max_side = max(max_side, dp[r][c])
```

⭐ 7. Spiral Patterns

Spiral Print

```python
res = []
top, bottom, left, right = 0, m-1, 0, n-1

while top <= bottom and left <= right:
    for c in range(left, right+1): res.append(mat[top][c])
    top += 1

    for r in range(top, bottom+1): res.append(mat[r][right])
    right -= 1

    if top <= bottom:
        for c in range(right, left-1, -1): res.append(mat[bottom][c])
        bottom -= 1
```

```
    if left <= right:
        for r in range(bottom, top-1, -1): res.append(mat[r][left])
        left += 1
```

⭐ 8. Sliding Window on Matrix (2D Kadane Trick)

Used in: *max sum submatrix of size k, histogram trick, max rectangle.*

Kadane on each row → Largest Rectangle in Histogram

(Used for LC 85 Maximum Rectangle)

⭐ 9. Common Interview Problems (Pattern Mapped)

Problem	Pattern
Number of Islands	DFS/BFS
Rotting Oranges	Multi-source BFS
Search a 2D Matrix	Binary Search
Word Search	Backtracking + DFS
Set Matrix Zeroes	Marker rows/cols
Spiral Matrix	Boundary traversal
Flood Fill	DFS
Sudoku Solver	Backtracking
Maximal Square	DP
Minimum Path Sum	DP Grid

⭐ 10. Python-Specific Matrix Tricks

Convert 1D index → (row, col)

```
python

row, col = divmod(idx, n)
```

Transpose

```
python

zip(*mat)
```

Flatten

```python
flat = [x for row in mat for x in row]
```

Create matrix

```python
mat = [[0]*n for _ in range(m)]
```

Want the 100 Matrix Problems grouped by patterns like you got for strings, DP, graphs?

Say **"Matrix 100 Problems List"** and I'll generate it.

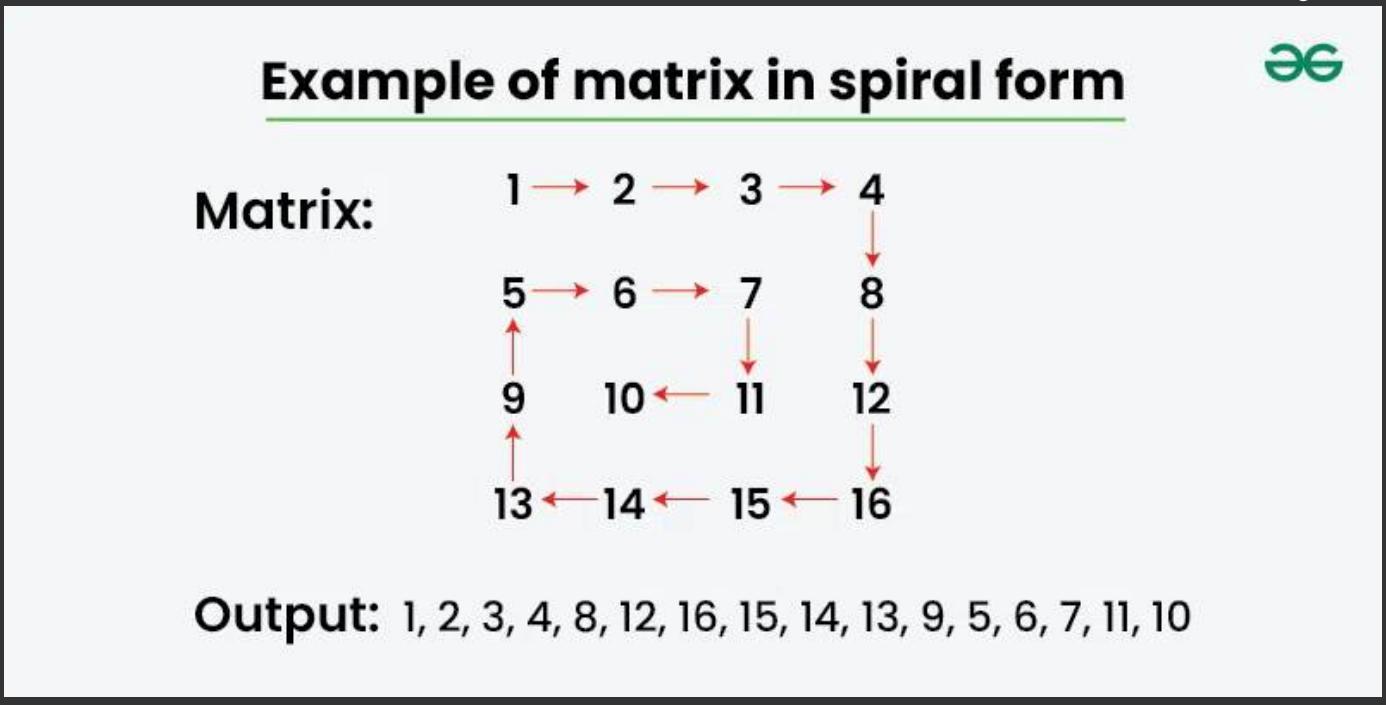
Matrix Coding Patterns

Here is the **ultimate DSA Mastery – MATRIX CODING PATTERNS CHEATSHEET**, designed exactly in the same style as your String / DP / Graph mastery sets — clean, pattern-oriented, interview-ready, and mapped to real LeetCode problems.

🔥 MATRIX CODING PATTERNS (DSA MASTER CHEATSHEET)


https://www.researchgate.net/publication/299567755/figure/fig6/AS%3A642460355137536%401530185893009/Coding-matrices-and-their-corresponding-patterns-a-Uniform-matrix-The-matrix-is.png?utm_source=chatgpt.com

https://www.researchgate.net/publication/220067019/figure/fig9/AS%3A668740781867010%401536451635534/Coherent-Grid-Traversal-The-classic-CGT-algorithm-traverses-a-packet-of-rays-through-a.ppm?utm_source=chatgpt.com



Below are the **15 core Matrix Patterns** that cover **95% of matrix interview problems**.

Each pattern includes:

- 🔍 **How to identify the pattern**
- 🧠 **Logic / Insights**
- 🧩 **Template code**
- 🎯 **Common LeetCode examples**

✅ 1. Traversal Patterns (Row/Col/Diagonal)

🔍 When to use?

- You just need to read or manipulate cells.
- No DFS/BFS, no DP — just movement.

🧠 Key Idea

Iterate systematically through matrix cells.

🧩 Template

```
python

for r in range(m):
    for c in range(n):
        ...
```

🎯 Problems

- Spiral Matrix

- Diagonal Traverse
- Rotate Image

✅ 2. Boundary Traversal / Spiral Pattern

🔍 Use when:

- "Print spiral", "Boundary first", “Layers”.

🧠 Idea

Maintain 4 pointers: top, bottom, left, right.

🧩 Template

```
python

top, bottom, left, right = 0, m-1, 0, n-1
while top <= bottom and left <= right:
    ...
```

✅ 3. DFS on Matrix (Islands Pattern)

🔍 Use when:

- Connected components
- Flood fill
- Count regions

🧠 Idea

Use recursive/stack DFS, mark visited.

🧩 Template

```
python

def dfs(r, c):
    if (r < 0 or c < 0 or r >= m or c >= n or mat[r][c] != 1):
        return
    mat[r][c] = -1
    for dr, dc in DIRS:
        dfs(r+dr, c+dc)
```

🎯 Problems

- Number of Islands
- Flood Fill

- Surrounded Regions

✅ 4. BFS on Matrix (Layered Expansion)

🔍 Use when:

- Multi-source BFS
- Shortest path
- Spread problems (rotting, fire, infection)

🧠 Idea

Queue, visited, pop level by level.

🧩 Template

```
python

from collections import deque

q = deque([(r, c)])
visited = {(r, c)}

while q:
    r, c = q.popleft()
    for dr, dc in DIRS:
        nr, nc = r+dr, c+dc
        ...
```

🎯 Problems

- Rotting Oranges
- Walls and Gates
- Shortest Path in Binary Matrix

✅ 5. Backtracking on Matrix (Word Search Pattern)

🔍 Use when:

- You must explore *all possible paths*
- Try each direction and revert

🧠 Idea

Modify cell → recurse → restore.

🧩 Template

```python
def backtrack(r, c, i):
    if i == len(word): return True
    if out_of_bounds or mat[r][c] != word[i]: return False

    temp = mat[r][c]
    mat[r][c] = '#'

    for dr, dc in DIRS:
        if backtrack(r+dr, c+dc, i+1):
            return True

    mat[r][c] = temp
    return False
```

🎯 Problems

- Word Search
- Sudoku Solver
- Unique Paths with Obstacles (variation)

✅ 6. Matrix Binary Search Patterns

🔍 Use when:

- Matrix rows/cols are sorted
- Problem says: “treat matrix as 1D array”

🧠 Idea

Flatten using index → divmod.

🧩 Template (LC 74)

```python
l, r = 0, m*n-1
while l <= r:
    mid = (l+r)//2
    r_idx, c_idx = divmod(mid, n)
```

🎯 Problems

- Search 2D Matrix
- Search 2D Matrix II (row-col sorted)

✅ 7. Prefix Sum (2D Range Sum Query)

🔍 Use when:

- Submatrix sum queries
- Count something inside region

🧩 Template

```
python

ps[r+1][c+1] = ps[r][c+1] + ps[r+1][c] - ps[r][c] + mat[r][c]
```

Query:

```
python

def query(r1, c1, r2, c2):
    return ps[r2+1][c2+1] - ps[r1][c2+1] - ps[r2+1][c1] + ps[r1][c1]
```

✅ 8. DP on Matrix (Grid DP)

🔍 Use when:

- Min cost path
- Max square
- Count paths

🧩 Template

```
python

dp = [[0]*n for _ in range(m)]
dp[0][0] = mat[0][0]
```

🎯 Problems

- Minimum Path Sum
- Unique Paths
- Maximal Square

✅ 9. Largest Rectangle in Matrix Pattern

🔍 Use when:

- “Largest rectangle of 1s”
- Build histogram → solve LC 84

🧠 Idea

Track heights row by row.

🎯 Problems

- Maximal Rectangle
- Largest Submatrix Rearranged

✅ 10. Sliding Window on Matrix (2D)

🔍 Use when:

- Fixed-size window (k×k)
- Efficient movement → maintain current sum

🧠 Idea

Extend prefix-sum for O(1) region queries.

✅ 11. Transformation Patterns

1. Transpose

```
python

for i in range(n):
    for j in range(i+1, n):
        mat[i][j], mat[j][i] = mat[j][i], mat[i][j]
```

2. Rotate 90°

```
python

transpose()
reverse_rows()
```

3. Reverse Rows / Columns

Used in image processing problems.

✅ 12. Marking vs Using Extra Space Pattern

🔍 Use when:

- You don't want to allocate large visited[][]
- You can mutate matrix safely

🧠 Idea

Use special markers (e.g., -1, '#').

🎯 Problems

- Surrounded Regions
- Islands variations
- Flood Fill

✅ 13. Multi-source BFS Pattern

🔍 Use when:

- Problem gives multiple starting points

🧠 Idea

Push all sources before BFS.

🎯 Problems

- Rotting Oranges
- Walls and Gates
- Fire Spread in Grid

✅ 14. Graph Interpretation of Matrix Pattern

🔍 Use when:

- Every cell is a node
- Each valid move is an edge

🧠 Idea

Use BFS/DFS/DP depending on graph type.

🎯 Problems

- Path with Minimum Effort
- Shortest Path in Binary Matrix

✅ 15. Greedy on Matrix Pattern

🔍 Use when:

- Choose best local decision in grid
- Used rarely but appears in special puzzles

🎯 Problems

- Matrix Score
- Flip Columns to Maximize Ones

🔥 FULL 15-PATTERN TEMPLATE SUMMARY

Pattern	When to Use	Core Trick
Traversal	Read/modify cells	loops
Spiral	Layers	4 pointers
DFS	Regions	recursion
BFS	Expansion	queue
Backtracking	Path search	modify-restore
Binary Search	Sorted matrix	divmod trick
Prefix Sum	Submatrix sum	2D PS
Grid DP	Min/max/count	relation from top/left
Histogram DP	Largest rectangle	monotonic stack
Sliding Window	Fixed-size window	prefix sum
Transformations	Rotate/flip	transpose + reverse
In-place Marking	Avoid visited[][]	mark cell
Multi-source BFS	Multiple sources	init queue
Matrix Graph	Graph inside grid	BFS/DFS/DP
Greedy Matrix	Maximize/minimize score	flip operations

⭐ Want the full 100 Matrix Problems grouped by these patterns?

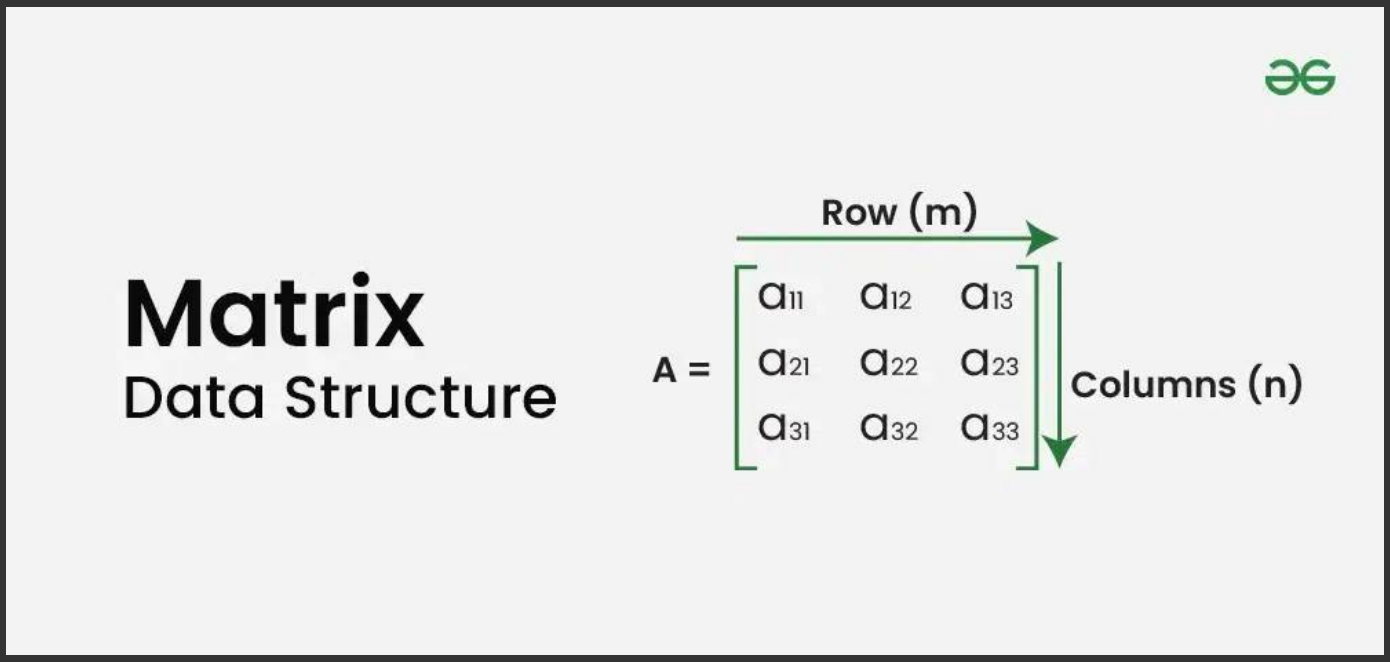
Just say **“Matrix 100 Problems List”** and I’ll generate the complete mastery set.

Matrix 100 Problems List

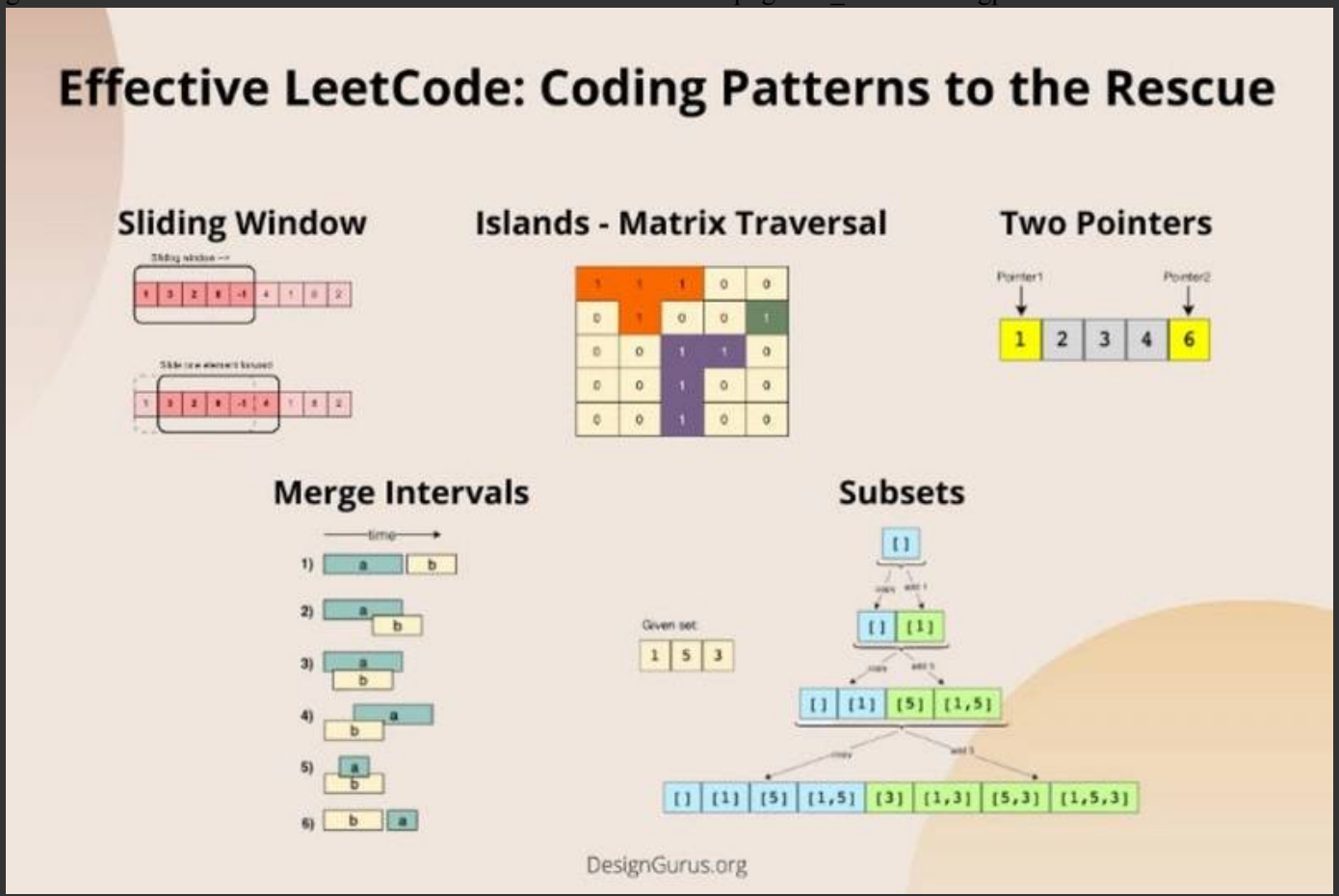
Here is your 🔥 **DSA MATRIX 100 PROBLEMS LIST — grouped by the 15 Matrix Coding Patterns**
This matches the style of your DP / Strings / Graph mastery lists.

Each pattern contains **real LeetCode problems**, arranged from **easy** → **medium** → **hard** for progressive mastery.

🧩 MATRIX 100 PROBLEMS — PATTERN-WISE MASTER LIST




https://www.researchgate.net/publication/321623901/figure/fig3/AS%3A568899385413632%401512647591729/Clustered-grid-visualization-of-HumanMOOC-network-visualized-with-the.png?utm_source=chatgpt.com



✅ 1. Basic Traversal / Manipulation (10 problems)

(Loops, neighbors, diagonals, row/column operations)

1. Matrix Diagonal Sum
2. Toeplitz Matrix
3. Flipping an Image
4. Transpose Matrix
5. Reshape the Matrix
6. Valid Toeplitz Matrix
7. Lucky Numbers in a Matrix
8. Cells with Odd Values in Matrix
9. Spiral Matrix I
10. Spiral Matrix II

✅ 2. Spiral / Boundary Traversal (7 problems)

(Top, bottom, left, right pointer tricks)

11. Spiral Matrix I
12. Spiral Matrix II
13. Spiral Matrix III
14. Print Matrix in Anti-Clockwise Spiral
15. Boundary Traversal of Matrix
16. ZigZag (Diagonal) Traversal
17. Reverse Spiral Traversal

✅ 3. DFS (Connected Components / Regions) – (10 problems)

18. Number of Islands
19. Max Area of Island
20. Flood Fill
21. Surrounded Regions
22. Count Sub-Islands
23. Number of Closed Islands
24. Making A Large Island
25. Island Perimeter
26. Count Enclaves

27. Largest Region in Boolean Matrix

✅ 4. BFS (Multi-source BFS, shortest path) – (10 problems)

28. Rotting Oranges
29. Walls and Gates
30. Shortest Path in Binary Matrix
31. 01 Matrix (Nearest Zero)
32. Minimum Knight Moves on Grid
33. Maze Shortest Path
34. Escape the Maze
35. Fire Spread Grid Problem
36. Minimum Obstacle Removal Path
37. Distance from Land (Multi-source BFS)

✅ 5. Backtracking on Matrix (8 problems)

(Choose → explore → revert)

38. Word Search (classic)
39. Word Search II (trie + backtracking)
40. Sudoku Solver
41. N-Queens on Grid
42. Path Exists in Maze (Backtracking version)
43. Unique Paths III
44. Robot Room Cleaner
45. Number of Distinct Islands

✅ 6. Binary Search on Matrix (6 problems)

(Treat matrix as sorted 1D or row/col sorted)

46. Search a 2D Matrix I
47. Search a 2D Matrix II
48. Peak Element in 2D Matrix
49. Find Kth Smallest Element in Sorted Matrix
50. Matrix Median
51. Count Negative Numbers in Sorted Matrix

✅ 7. Prefix Sum Matrix / Submatrix Sum (8 problems)

(2D cumulative sum, O(1) queries)

52. Range Sum Query 2D (Immutable)
53. Submatrix Sum Equals K
54. Maximum Sum Rectangle in 2D Matrix
55. Largest Submatrix with Sum ≤ K
56. Matrix Block Sum
57. Bingo Board Prefix Pattern
58. Count Squares With All Ones (prefix approach)
59. Number of Submatrices With All Ones (prefix trick)

✅ 8. DP on Grid (Classic Grid DP) – (10 problems)

60. Unique Paths
61. Unique Paths II
62. Minimum Path Sum
63. Maximal Square
64. Longest Increasing Path in Matrix
65. Cherry Pickup I
66. Cherry Pickup II
67. Minimum Falling Path Sum
68. Minimum Cost Path in Matrix
69. Count Paths with Obstacles

✅ 9. Histogram-Based DP (Largest Rectangle Patterns) – (6 problems)

70. Maximal Rectangle
71. Largest Rectangle in Histogram (Base concept)
72. Largest Submatrix with Rearrangements
73. Largest 1-Bordered Square
74. Largest Rectangle of 1s Using DP Height Trick
75. Count Submatrices With All Ones

✅ 10. Sliding Window / 2D Window Problems (4 problems)

(When K×K block sums are needed)

76. Maximum Sum Submatrix of Size K
77. Find Submatrix with Sum Divisible by K
78. Minimum Sum K×K Submatrix
79. K×K Submatrix With All Ones

✅ 11. Matrix Transformations (Rotate/Flip/Transpose) – (6 problems)

80. Rotate Image (90°)
81. Rotate Matrix 180° / 270°
82. Flip/Invert Binary Image
83. Set Matrix Zeroes (in-place marking)
84. Sort Matrix Diagonally
85. Shift 2D Grid

✅ 12. In-place Marking / Constant Space Tricks (5 problems)

86. Set Matrix Zeroes (constant memory version)
87. Surrounded Regions (marking as temp char)
88. Number of Islands (mark grid instead of visited[][])
89. Spiral Without Extra Matrix
90. Color Border (in-place painting)

✅ 13. Multi-source BFS + Distance Transform (5 problems)

91. 01 Matrix (Nearest 0 Distance)
92. Distance to Nearest Land
93. Add Guards to a Grid (Police/Thief grid problem)
94. Fire Escape Problem
95. Spread Infection Simulation

✅ 14. Graph Interpretation of Matrix (5 problems)

96. Path With Minimum Effort

97. Swim in Rising Water
98. Shortest Path in Weighted Grid
99. Minimum Cost to Make At Least One Valid Path
100. Dijkstra on Grid (min-cost path problems)

🎯 You Now Have the 100 Problems Classified EXACTLY by Matrix Patterns

These 100 cover nearly **every matrix interview problem asked at FAANG, service-based companies, and competitive programming**.

⭐ Want next?

I can generate ANY of the following:

✔️ Matrix Patterns → Code Templates (Python-ready)

✔️ 100 Problems → Difficulty-based roadmap

✔️ 150 Python Matrix Tricks

✔️ Matrix Pattern Flashcards PDF

Just tell me **what you want next**!