



Trishanth Naidu

Software Developer | Founder of Peppy UI | Author of Rootz JS
Educator at Relevel

Innovator | Fitness freak | Painter | Chef

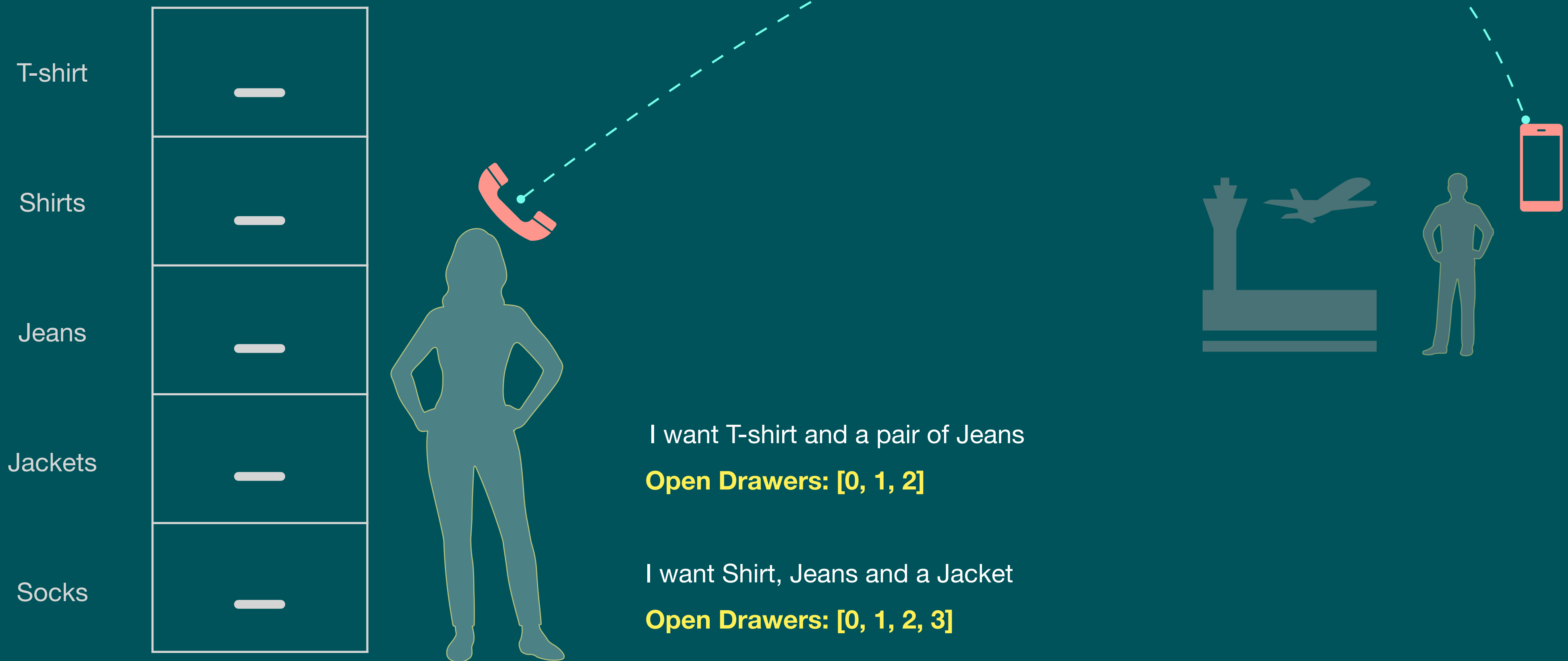
You can Follow me on



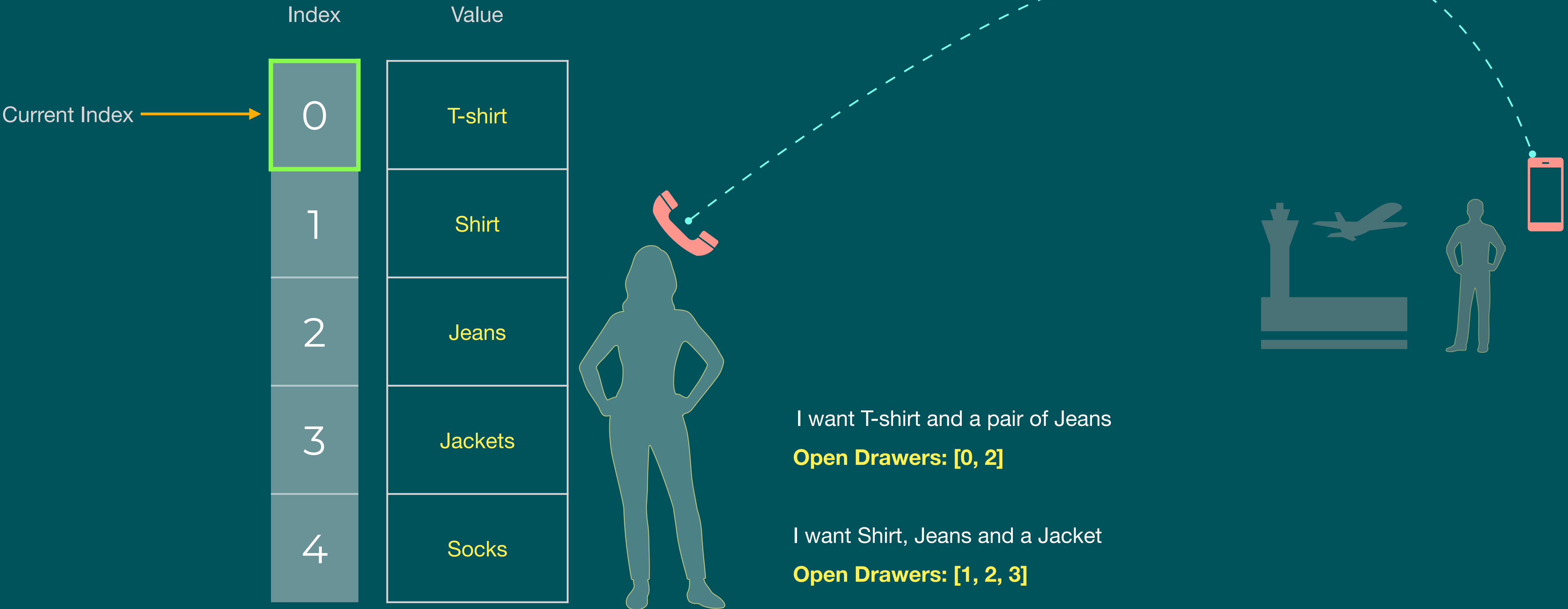
Problems are never boring, lack of understanding them makes them so.

- Trishanth Naidu

What is 1D Array ?



What is 1D Array ?



What is 2D Array ?

Rows

1	2	3	4
5	6	7	8
9	1	2	3

Columns

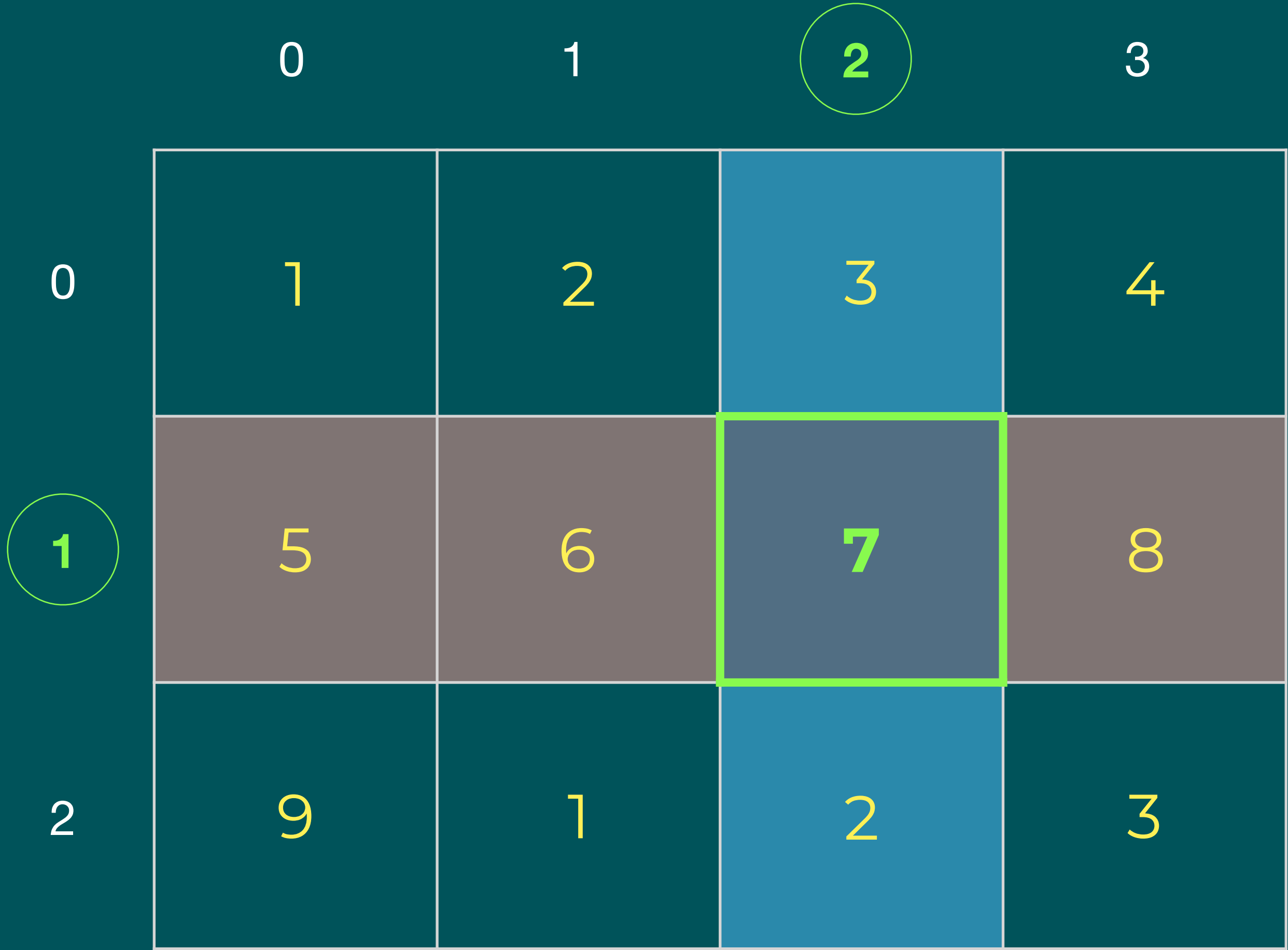
1	2	3	4
5	6	7	8
9	1	2	3

Positioning of cells in 2D Array ?

0	1	2	3	4
1	5	6	7	8
2	9	1	2	3

	0	1	2	3
	1	2	3	4
	5	6	7	8
	9	1	2	3

Positioning of cells in 2D Array ?



2D Array with JavaScript syntax

```
[  
  1,  
  2,  
  3  
]
```

0
1
2

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	1	2	3

0
1
2

```
[  
  [(0, 0), (0, 1), (0, 2), (0, 3)]  
  [(1, 0), (1, 1), (1, 2), (1, 3)]  
  [(2, 0), (2, 1), (2, 2), (2, 3)]  
]
```


2D Array with JavaScript syntax

Push	Adds element to the end of the array
Pop	Removes element from the end of the array
Shift	Removes element from the start of the array
Unshift	Adds element to the start of the array
Splice	Removes elements from the start index and number elements specified by deleteCount and add new elements at the end of the array
Slice	Removes elements from the start of the array as per distance mentioned
Sort	Sorts the order of elements within array

Remember

If you want to print **Rows** you would need to traverse through the **Columns**.

And

If you want to print **Columns** you would need to traverse through the **Rows**.

Print matrix in spiral form

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

- 1. Keep the row[rowStartIndex] constant, traverse through the column[colStartIndex -> colEndIndex]
- 2. Increase the rowStartIndex by 1

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

- 1. Keep the column[colEndIndex] constant, traverse through the row[rowStartIndex -> rowEndIndex]
- 2. Decrease the colEndIndex by 1

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

- 1. Keep the row[rowEndIndex] constant, traverse through the column[colEndIndex -> colStartIndex]
- 2. Decrease the rowEndIndex by 1

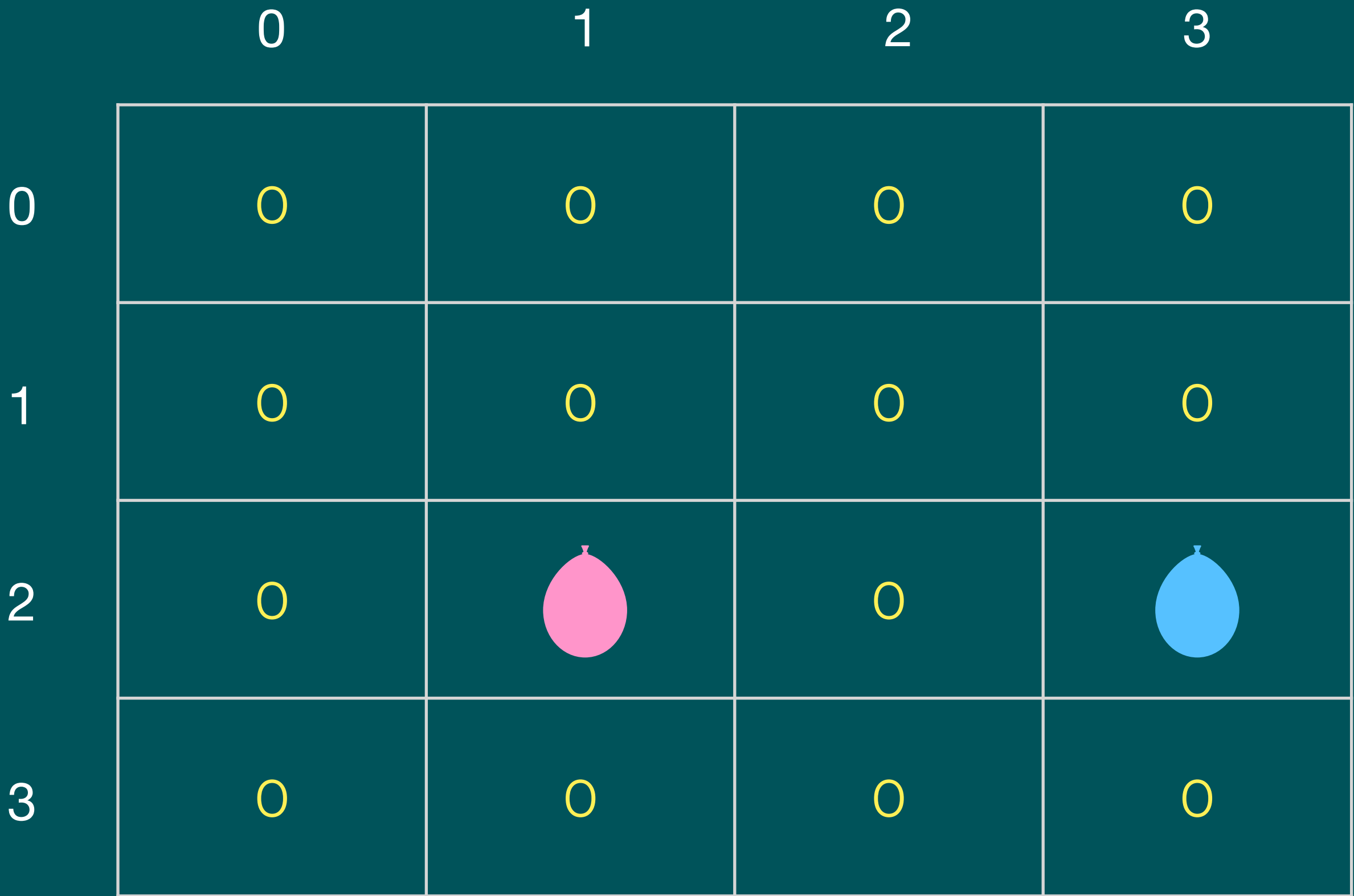
	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

- 1. Keep the column[colStartIndex] constant, traverse through the row[rowEndIndex -> rowStartIndex]
- 2. Increase the colStartIndex by 1



Explode bombs problem

Input:
bombs [[2, 3] [2, 1]]
nRows = 4
nCols = 4

Output:
[
 [0, 0, 0, 0],
 [1, 1, 2, 1],
 [1, -1, 2, -1],
 [1, 1, 2, 1],
]



Rotate matrix by 90 deg

	0	1	2	3
0	0	0	0	0
1	1	1	2	1
2	1		2	
3	1	1	2	1

- 1. Fill the matrix with 0
- 2. Pick the row and column of the bomb placed
- 3. Make it -1, as placing the bomb there
- 4. Loop from 1 row prior to the row of bomb placed to 1 row next of the bomb placed.
- 5. Loop from 1 col prior to the col of bomb placed to 1 col next of the bomb placed.
- 6. Check if the current location is between row 0 -> 3 AND column 0 -> 3 AND current location is not the bomb's location.
- 7. Increase the location count by 1

Rotate matrix by 90 deg

N is the length of the row and column of the matrix, 3

If the rows are termed as i in a loop running from 0 -> 2. Then one equation for all the row values at the extreme edges / corners would be $N - 1 - i$

Same for Column will be $N - 1 - j$

		j		
		0	1	2
i	0	1	2	3
	1	4	5	6
	2	7	8	9

Rotate matrix by 90 deg

1	2	3
4	5	6
7	8	9

		1

```
var result = [];  
function rotate(mat, N) {  
    for(i =0; i<N; i++) {  
        result[i] = [];  
        for(j = N-1; j >= 0; j--) {  
            result[i][N - 1 - j] = mat[j][i];  
        }  
    }  
    return result;  
}  
  
rotate([[1,2,3], [4,5,6], [7,8,9]], 3)
```

Print in wave form

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	1	2	3

```
var result = [];  
function rotate(mat, r, c) {  
  for(i = 0; i < c; i++) {  
    if(i % 2 === 0) {  
      for(j = 0; j < r; j++) {  
        result.push(mat[j][i]);  
      }  
    } else {  
      for(j = r-1; j >= 0; j--) {  
        result.push(mat[j][i]);  
      }  
    }  
  }  
  return result;  
}  
  
rotate([[1,2,3,4], [5,6,7,8], [9,1,2,3]], 3, 4)
```


Transpose matrix

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

	0	1	2
0	1	4	7
1	2	5	8
2	3	6	9

0,0	0,1	0,2
0,0	1,0	2,0
1,0	1,1	1,2
0,1	1,1	2,1
2,0	2,1	2,2
0,2	1,2	2,2

```
var rowLen = 3;
var colLen = 3;
var mat = [[1,2,3], [4,5,6], [7,8,9]];
var result = new Array(colLen);

for(i = 0; i < colLen; i++) {
    result[i] = new Array(rowLen);
}

function transpose(mat, r, c) {
    for(i = 0; i < r; i++) {
        for(j = 0; j < c; j++) {
            result[j][i] = (mat[i][j]);
        }
    }

    return result;
}

transpose(mat, rowLen, colLen);
```

Multiply matrix

A	B
C	D

X

E	F
G	H

=

AE + BG	AF + BH
CE + DG	CF + DH

Multiply matrix

A

1	1
2	2

B

1	1
2	2

C

1	0
0	0

A

1	1
2	2

B

1	1
2	2

C

3	1
0	0

1	1
2	2

1	1
2	2

1 + 2	0
0	0

1	1
2	2

1	1
2	2

3	1 + 2
0	0

A

1	1
2	2

B

1	1
2	2

C

3	3
2	0

A

1	1
2	2

B

1	1
2	2

C

3	3
4	2

1	1
2	2

1	1
2	2

3	3
2 + 4	0

1	1
2	2

1	1
2	2

3	3
6	2 + 4

Multiply matrix

```
function multiplyMatrices(A, B, r1, c1, r2, c2, C) {
    for(var i = 0; i < r1; i++) {
        for(var j = 0; j < c2; j++) {
            C[i][j] = 0;
            for(var k = 0; k < c1; k++) {
                C[i][j] = C[i][j] + A[i][k]*B[k][j];
            }
        }
    }
    return C;
}
```

```
multiplyMatrices([[1,1], [2,2]], [[1,1], [2,2]], 2, 2,
2, 2, [[],[]])
```

```
let A = [
    [ 1, 1, 1 ],
    [ 2, 2, 2 ],
    [ 3, 3, 3 ],
    [ 4, 4, 4 ],
];
```

```
let B = [
    [ 1, 1, 1, 1 ],
    [ 2, 2, 2, 2 ],
    [ 3, 3, 3, 3 ]
];
```

```
multiplyMatrices(A, B, 4, 3, 3, 4, [[],[], [], []])
```