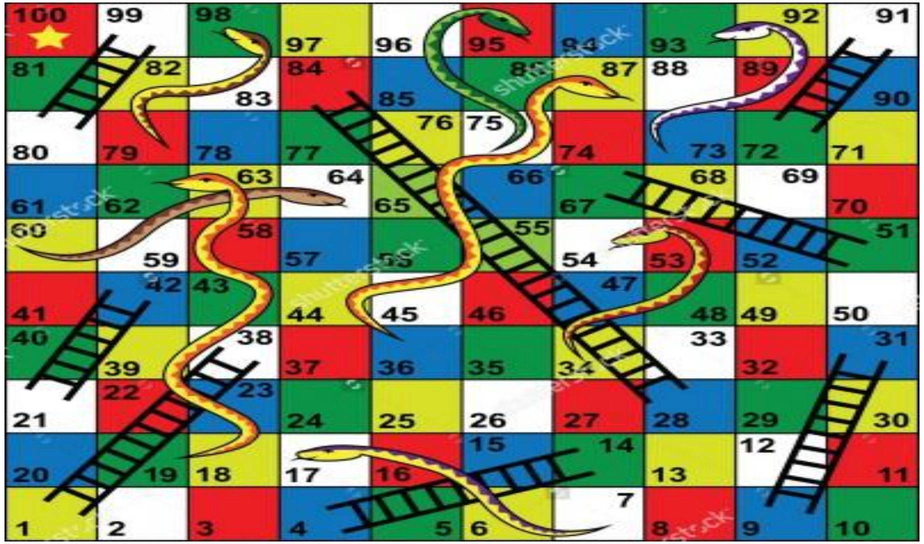
2D Arrays and Problem Solving in JS

Relevel
by Unacademy

2D Arrays Introduction

Snake and Ladder game matrix with say N rows and M columns



What is a 2D Array?

Can be visualised as a grid of n rows and m columns

```
[
     [1, 2, 3, 4]
     [5, 6, 7, 8]
     [9, 1, 2, 3]
]
```

Corresponding position matrix looks like

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

How is a 2D array stored in memory?

- Stored in row-wise order
- A matrix stored at location 0 with c columns and consisting of elements of size B each has it’s (i, j)th element stored at memory location:

$$(i*c + j)*B$$

2D array programming in JAVASCRIPT

```
var n = 4, m = 5;
var matrix = new Array(n);

for (var i = 0; i < matrix.length; i++)
{
    matrix[i] = new Array(m);
}
var cnt = 0;

for (var i = 0; i < n; i++) {
    for (var j = 0; j < m; j++) {
        matrix[i][j] = cnt++;
    }
}
```

```
for (var i = 0; i < n; i++) {
    for (var j = 0; j < m; j++)  {
        console.log(matrix[i][j]
+ " ");
    }
    console.log('\n');
}
```

Advanced 2D array programming in JAVASCRIPT

```
const m = 4;
const  n = 5;
let arr = Array.from(Array(m), ()  =>
new Array(n));
console.log(arr); //  Output: [ [ <n
empty items, here 5> ], [ <5 empty
items> ], [ <5 empty items> ], [ <5
empty items> ] ]
```

```
const m = 4;
const  n = 5;
let arr = Array(m).fill().map(()  =>
Array(n));
console.log(arr);  //  Output: [ [ <5
empty items> ], [ <5 empty items> ],
[ <5 empty items> ], , [ <5 empty
items> ] ]
```

Important Functions of 2D array

- **push(value):** adds an element to the end of the array and returns the new size of the array.
- **pop():** returns and removes last element of the array
- **splice(start, deleteCount, item1, item2, itemN):**
  - remove deleteCount elements from the index start of the array and add new items.

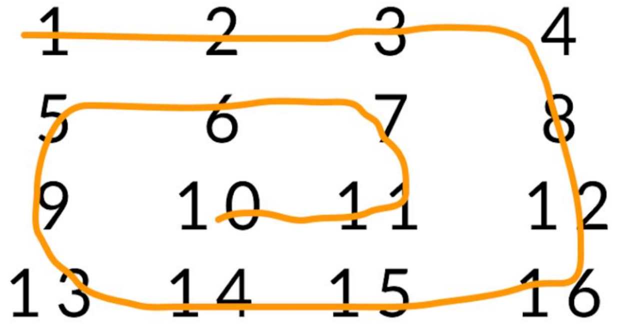
Continued....(forEach loop in JS)

```
// iterating on employee data 2D array
employeesData.forEach((employee) => {
    employee.forEach((data) => {
        console.log(data);
    });
});
```

Multi-dimensional Arrays Problem Solving

Print Matrix in Spiral form

Given matrix $\longrightarrow$



$$\begin{matrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{matrix}$$

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

Spiral Matrix Logic

```
In every rotation of a spiral, we know that we will follow four
directions, so we can print the cell values in these 4 directions
using 4 for loops, inside an outer loop, which runs the same 4 for
loops, analogous to every rotation.
```

Explode Bombs Problem.

- Given a matrix. There are bombs placed at some cells denoted by value -1. So, wherever a bomb is placed, the cells around it explodes which means there value gets incremented by 1 except if that is a bomb too. Initial value of all empty cells is assumed to be 0. So, print the final state of the matrix after all bombs explode.

  Algorithm:

  1. Iterate on the bombs array.
  2. Set the cells with bomb to -1 in the output matrix initialised to 0.
  3. Then, visit all the cells around the bomb and increment them by 1 if it does not contain a bomb already.
  4. Print this output matrix.

Rotate a square matrix by 90 degrees in a clockwise direction without using any extra space.

Logic:

- Square Cycle: In every iteration, the elements **mat[i][j]** top leftmost element, **mat[j][n-1-i]** top rightmost element, **mat[n-1-i][n-1-j]** bottom rightmost element, **mat[n-1-j][i]** bottom leftmost element, are getting rotated or swapped with their neighbors.

- There will be floor(N/2) square cycles in an NxN matrix. In each square cycle, we will swap corresponding elements while rotating in a clockwise direction.

Print Matrix in wave form.

Logic:

- Print the first column in downward direction, then the second column in upward direction and so on.
- Increment to next column whenever you reach the extremes of the current column, while printing the matrix in column-wise fashion.

Transpose of a matrix.

- Transpose of a matrix means exchanging the row values with corresponding columns and column values with corresponding rows.
- As we can see, we have to use swap operation here to exchange the rows with columns or just place the element at (j, i)th positioned element in the matrix to (i, j)th positioned element in the output matrix.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Input

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

Output

Multiply two matrices

$$\underset{A}{\begin{bmatrix} a & b \\ c & d \end{bmatrix}} \times \underset{B}{\begin{bmatrix} e & f \\ g & h \end{bmatrix}} = \underset{C}{\begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}}$$

A, B and C are square metrices of size N x N
a, b, c and d are submatrices of A, of size N/2 x N/2
e, f, g and h are submatrices of B, of size N/2 x N/2

Practice/HW

- Write an efficient algorithm to search a value in a 2D matrix in which the rows are in sorted order and the first integer of each row is greater than the last integer of the previous row.

- Find the row index which has maximum no. of unique elements in a matrix efficiently.

Next Class

- In the next class, we would be looking into doubt solving and some problems on searching algorithms.

THANK YOU