06/03/2024

PAGE No.
DATE / /

Day 9 of DSA.

Task.

Checkbox Matrix (2D Arrays)

- [✓] Matrix operations : Addition, Subtraction
- [✓] Basic Matrix Multiplication : Naive Matrix multiplication
- [✓] Implement addition, subtraction of matrices
- [✓] Implement naive Matrix multiplication

※ 2·D Array:-

→ Declaration:-

```
int arr[i][j];

int arr[2][2] = {1, 2, 3, 4};

int arr[2][2] = {{1, 2}, {3, 4}};
```

col

row	0	1
0	1	2
1	3	4.

→ Input :-

```
For (int row=0; row<n; row++)
{
    For( int col =0; col<n; col++)
    {
        cin>> arr[row][col];
    }
}
```

→ Output :-

```
Keep 2 For loop Same as it is
{
  {
    cout << arr[row][col] << " ";
  }
  cout << endl;
}
```

PAGE No.	
DATE	/ /

* Row-wise Input:-

```
For(int row=0; row<n; row++)
{
    For(int col=0; col<n; col++)
    {
        cin>>arr[row][col];
    };
}
```

* Col-wise input:-

```
For( int col=0; col<n; col++)
{
    For( int row=0; row<n; row++)
    {
        cin >> arr[row][col];
    }
}
```

* Linear search:-

```
For( int row=0; row<n; row++)
{
    For( int col=0; col<n; col++)
    {
        if (arr[row][col] == target)
            return 1;
    }
}
```

PAGE No.
DATE / /

* Row - wise sum.

```
            int sum = 0;
    For( int row = 0; row < n; row ++)
    {
        for( int col = 0; col < n; col++)
        {
            sum += arr[row][col];
        }
    }
```

* col - wise sum :

```
            int sum = 0;
    For( int col = 0; col < n; col ++)
    {
        For( int row = 0; row < n; row ++)
        {
            sum += arr[row][col];
        }
    }
```

* Largest Row Sum :-

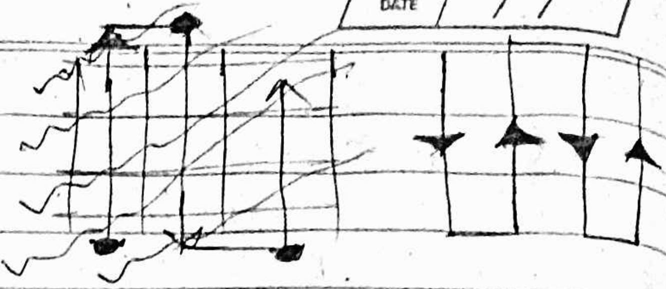
```
    For( int row = 0; row < n; row ++)
    {
        For( int col = 0; col < n; col ++)
        {
            sum += arr[row][col];
        }
        if (sum > maxi)
        { maxi = sum; ✓
          rowindex = row; ✓
        }
    }
```

```
maxi = Int_MIN
rowindex = -1
```

PAGE No.

DATE / /

① Print wave :-

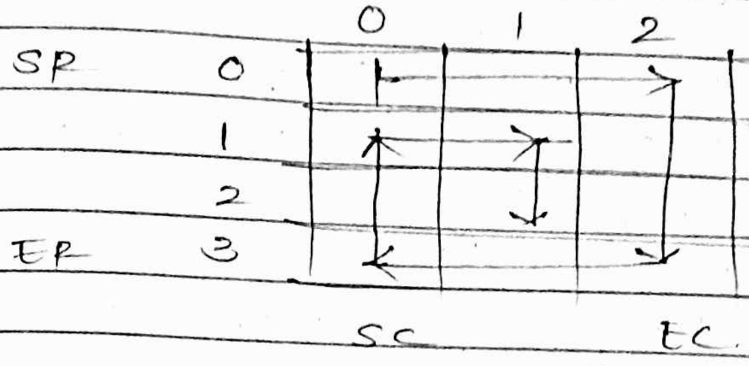


```
For( int col=0; col<n; col++)
{
    if( col&1)          mean if it is odd
                        Bottom to top
    {
     For (int row=n; row>=0; row--)
    {
      cout<< ~~row~~ arr[row] [col]<<" ";
    }
    }else {             mean it is over
                        Top to Bottom
     For( int row =0; row<n; row++)
    {
     cout << ~~arr~~arr[row][col] ]<< " ";
    }
}
```

Time Complexity : O(n×m)

$\boxed{\text{Time complexity , } O(n*m)}$

(2) Spiral Matrix.



int count = 0
int total = $\underbrace{\text{row} ~~*~~ \text{col}}_{\text{length}}$;

startingRow = 0
starting Col = 0
Ending Row = Row-1;
ending col = col-1;

while (count < total)
{

(1)
For(int index = starting ~~Row~~ Col ; index <= endingCol && count < total index++
{
cout << arr[~~index~~ SR][~~starting~~ SC];
}
startingRow ++

(2)
For(int index = SR ; count < total && index <= SR, index ++)
{
Cout << arr[index][ending col];
}
ending Col --;

(3)
For(int index = endingcol ; count < total && index >= startcol; index --)
{ cout << arr[endR][index];
}
ending Row --;

(4)
For(int index = ending Row ; i >= startingRow && count < total index--)
{
Cout << arr[index) [startingcol]);
}

PAGE No.
DATE / /

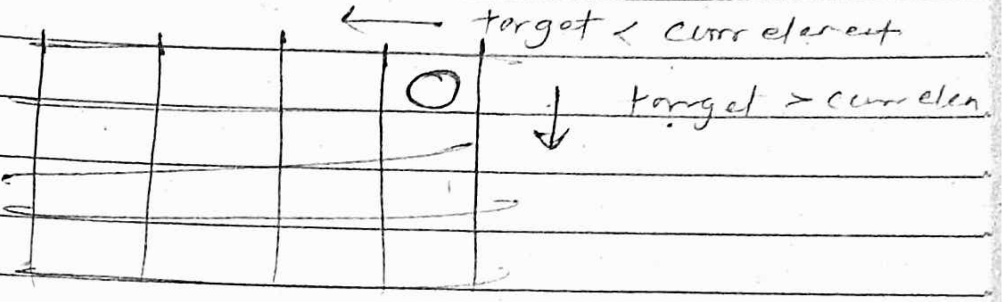
③ Search in 2D - Matrix Level ①

```
int row = matrix[].size();
int col = matrix[0].size();

int start = 0;
int end = col - 1;
         *
        row

while ( start <= end )
{
   int mid = start + (end-start)/2;

   int element = arr[mid/col][mid%col];
                          target
   if ( element == ~~mid~~ )
   {
       return 1;
   }

   if ( element > target )
   {
       end = mid-1;
   }
   else
   {
       start = mid+1;
   }
}
```

| Time complexity : O(log(r*c)) |

←
①

④ Search in 2D-Matrix II :-



```
int rowI= 0
int colI= col-1;
          size

while( ~~row < col~~ rowI <row.size() &&
                     colI >=0)
{
   int element = matrix [rowIndex] [colIndex];

  if (element == target)
      return 1;

  if ( element < target )
         ~~col--~~ row++;
  else
         col--;

}
```

$\boxed{\text{Time Complexity} = O(\log(n*m))}$

PAGE No.
DATE / /

(5) Rotate 2D Array by 90 deg.

$$\begin{array}{cccc} & \longrightarrow & & \\ \uparrow & 1 & 2 & 3 \\ & 4 & 5 & 6 \\ & 7 & 8 & 9 \end{array}$$

$$\downarrow$$

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

य हर ~~Row~~ Col को मुड़े Row wise print करना है।

```
For(   col = 0 → n )
{
    For(   Row = n → 0 )
    {
        cout << arr [Row][col];
    }
}
```
✓