

Lecture 17

2D Array - In-Class Questions

Get row with max number of 1s

```
public static void main (String[] args) {  
    Scanner input = new Scanner(System.in);  
    int rows = input.nextInt();  
    int cols = input.nextInt();  
  
    int[][] matrix = new int[rows][cols];  
    for(int i = 0; i < rows; i++) {  
        for(int j = 0; j < cols; j++) {  
            matrix[i][j] = input.nextInt();  
        }  
    }  
  
    System.out.println(getRowWithMaximumOnes(matrix));  
}
```

Get row with max number of 1s

```
public static int getRowWithMaximumOnes( int[][] matrix) {  
    int rows = matrix.length;  
    int cols = matrix[0].length;  
  
    int rowWithMaxOnes = -1;  
    int currentMaxNumberOfOnes = 0;  
    int noOfOnes = 0;  
    for(int i = 0; i < rows; i++) { // moving from one row to next  
        noOfOnes = 0;  
        for(int j = 0; j < cols; j++) { // moving from one columns to next  
            if(matrix[i][j] == 1) { // we calculate the number of 1s in the current row  
                noOfOnes++;  
            }  
        }  
  
        if(noOfOnes > currentMaxNumberOfOnes) { // if current row has more 1s than previous maximum row  
            currentMaxNumberOfOnes = noOfOnes;  
            rowWithMaxOnes = i;  
        }  
    }  
  
    return rowWithMaxOnes;  
}
```

Diagonal Sum

```
cols = 1 - i
cols = 4;
firstIndex = 0;
lastIndex = cols - 1
row 0 , secondaryDiagonalColumn = 4 - 1 - 0 = 3
row 1 , secondaryDiagonalColumn = 4 - 1 - 1 = 2
row 2 , secondaryDiagonalColumn = 4 - 1 - 2 = 1
row 3 , secondaryDiagonalColumn = 4 - 1 - 3 = 0
```

rows

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

```
1 1 1 1
2 2 2 2
3 3 3 3
4 4 5 5
```

```
1 + 2 + 3 + 5 = 11
```

```
1 + 2 + 3 + 4 = 10
```

Diagonal Sum

```
public static void main (String[] args) {  
    Scanner input =new Scanner(System.in);  
    int rows = input.nextInt();  
    int cols = rows;  
  
    int[][] matrix = new int[rows][cols];  
  
    for(int i = 0; i < rows; i++) {  
        for(int j = 0; j < cols; j++) {  
            matrix[i][j] = input.nextInt();  
        }  
    }  
  
    getSumOfPrimarySecondaryDiagonals(matrix);  
}
```

Diagonal Sum

```
public static void getSumOfPrimarySecondaryDiagonals(int[][] matrix) {  
    int rows = matrix.length;  
    int cols = matrix[0].length;  
  
    int primaryDiagonalSum = 0;  
    for(int i = 0; i < rows; i++) {  
        primaryDiagonalSum += matrix[i][i];  
    }  
  
    int secondaryDiagonalSum = 0;  
    for(int i = 0; i < rows; i++) {  
        secondaryDiagonalSum += matrix[i][cols - 1 - i];  
    }  
  
    System.out.println(primaryDiagonalSum + " " + secondaryDiagonalSum);  
}
```

Rotate Matrix

```
public static void main(String[] args) {  
    int[][] matrix = {  
        { 1,2,3,4},  
        { 5,6, 7,8},  
        { 9,10, 11, 12},  
        { 13,14,15,16}  
    };  
  
    print90Degrees(matrix);  
}  
  
public static void print90Degrees(int[][] matrix) {  
    int rows = matrix.length;  
  
    for(int i = 0; i < rows; i++) { // This decides which column to print  
        for(int j = rows - 1; j >= 0; j--) { // This decides which row to print  
            System.out.println( "row = " + j + " and column = " + i);  
            // System.out.print(matrix[j][i] + " ");  
        }  
        System.out.println();  
    }  
}
```