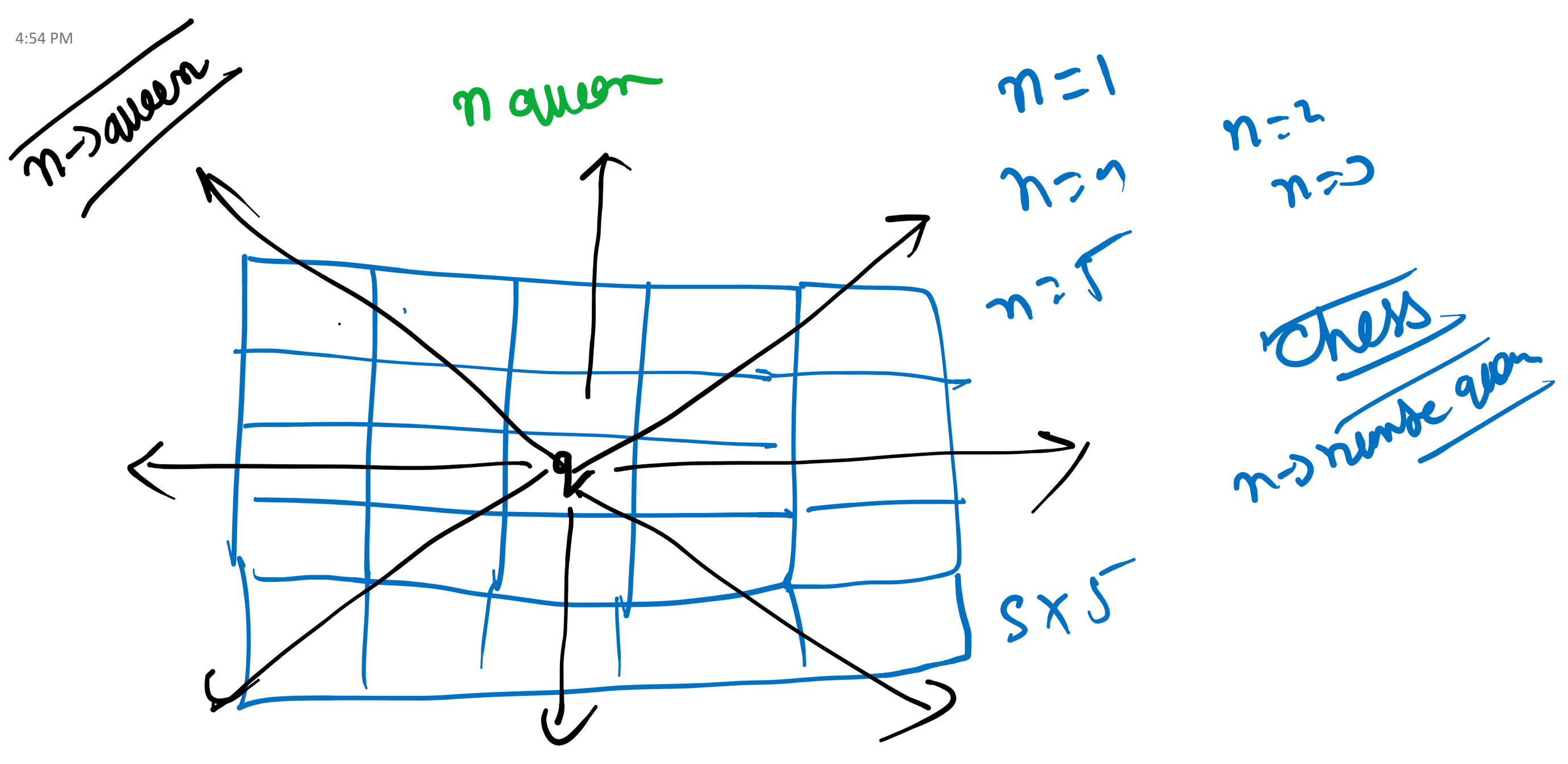
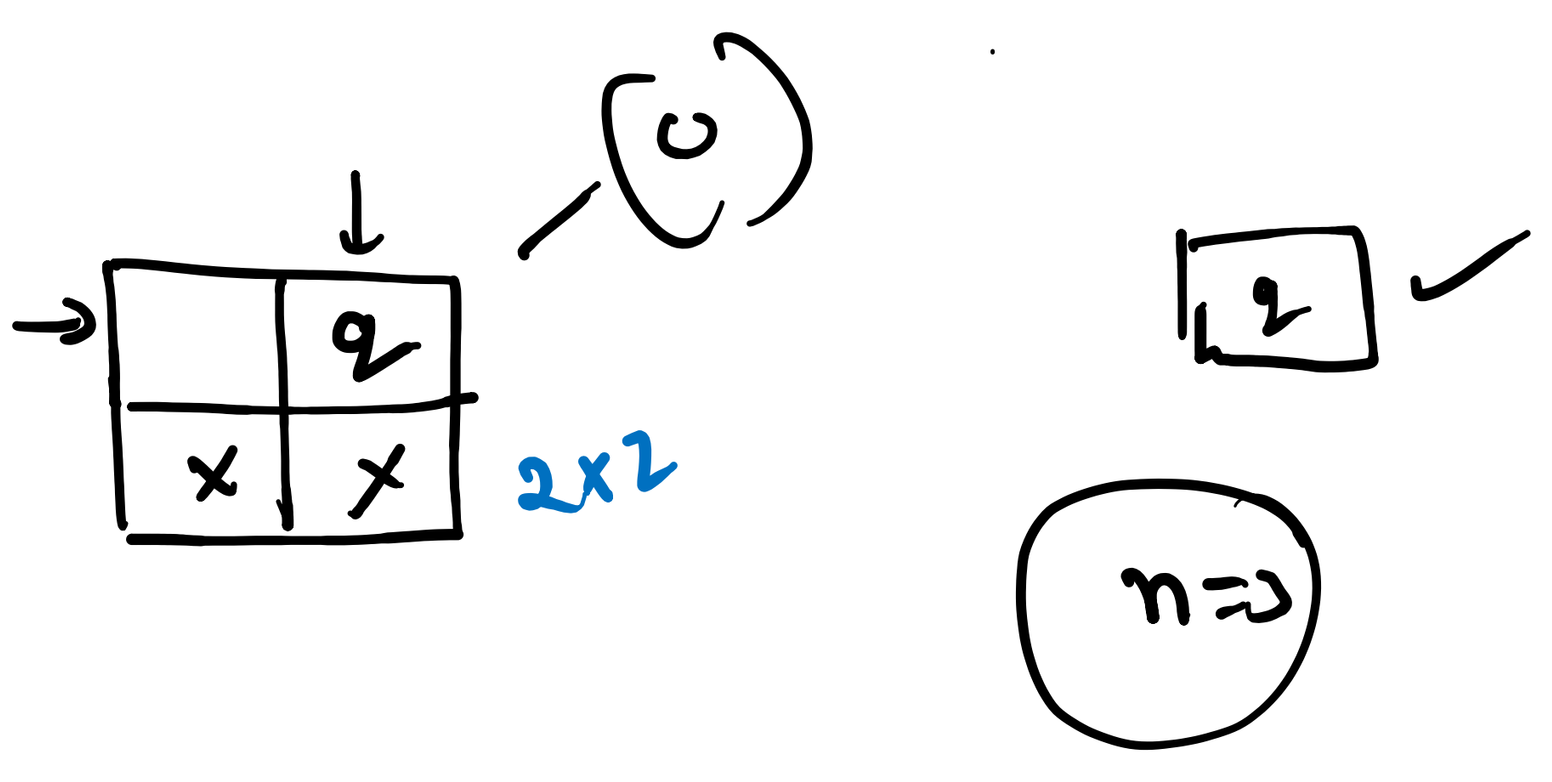
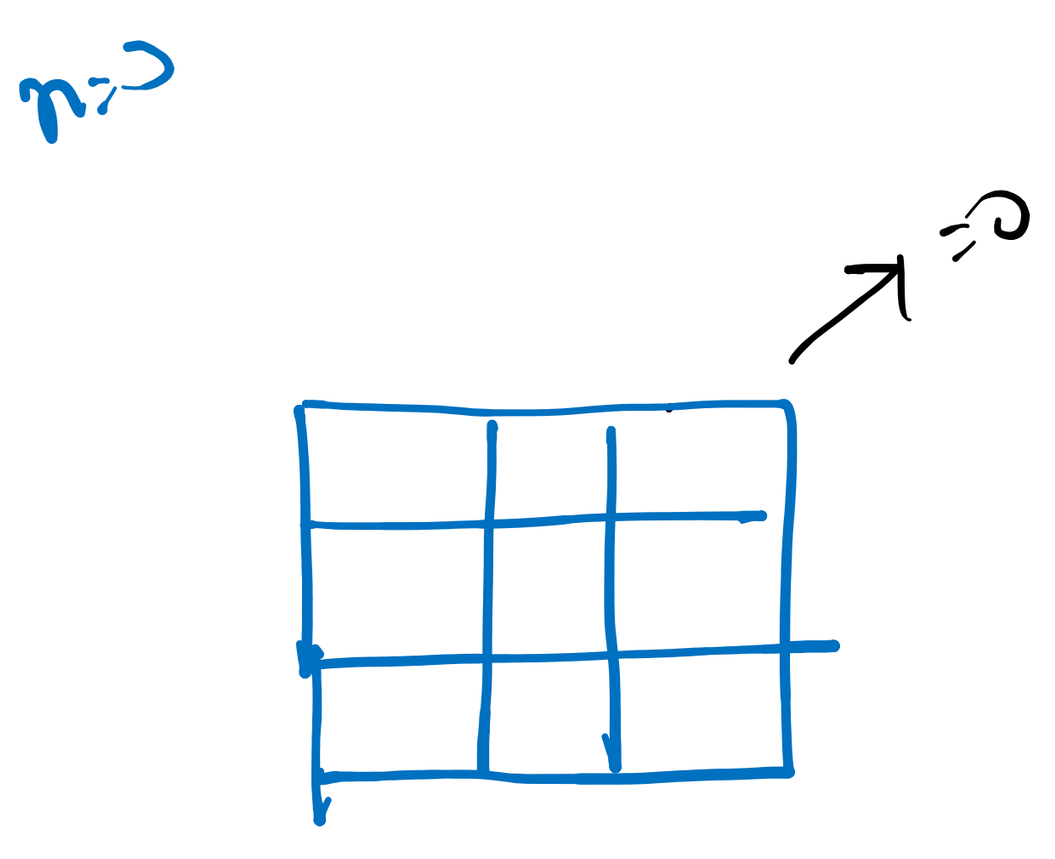
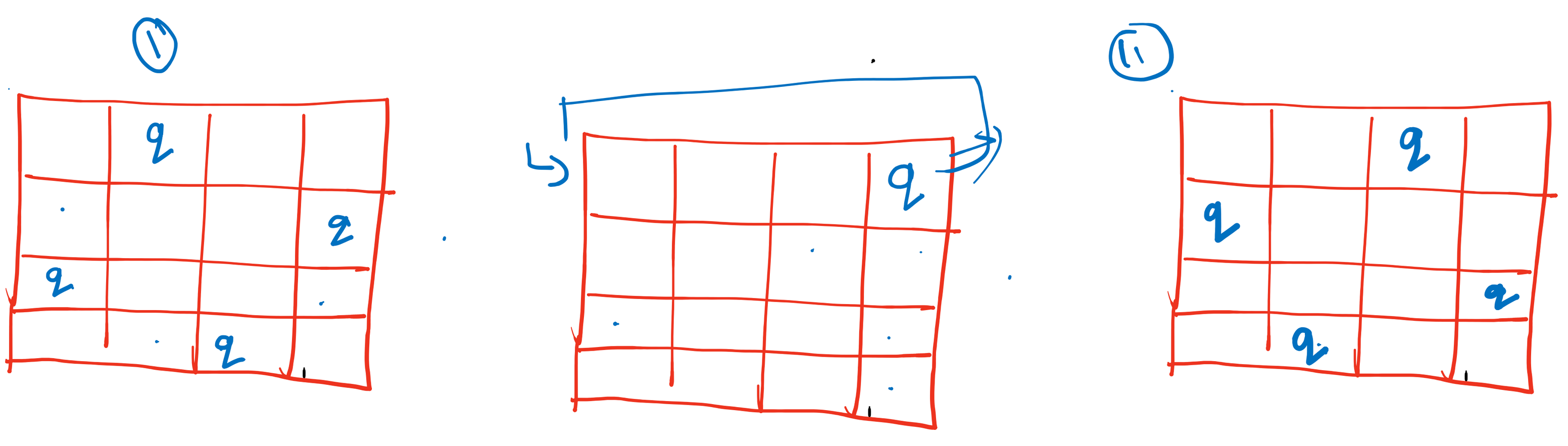
Lec-21

27/07/24 4:54 PM



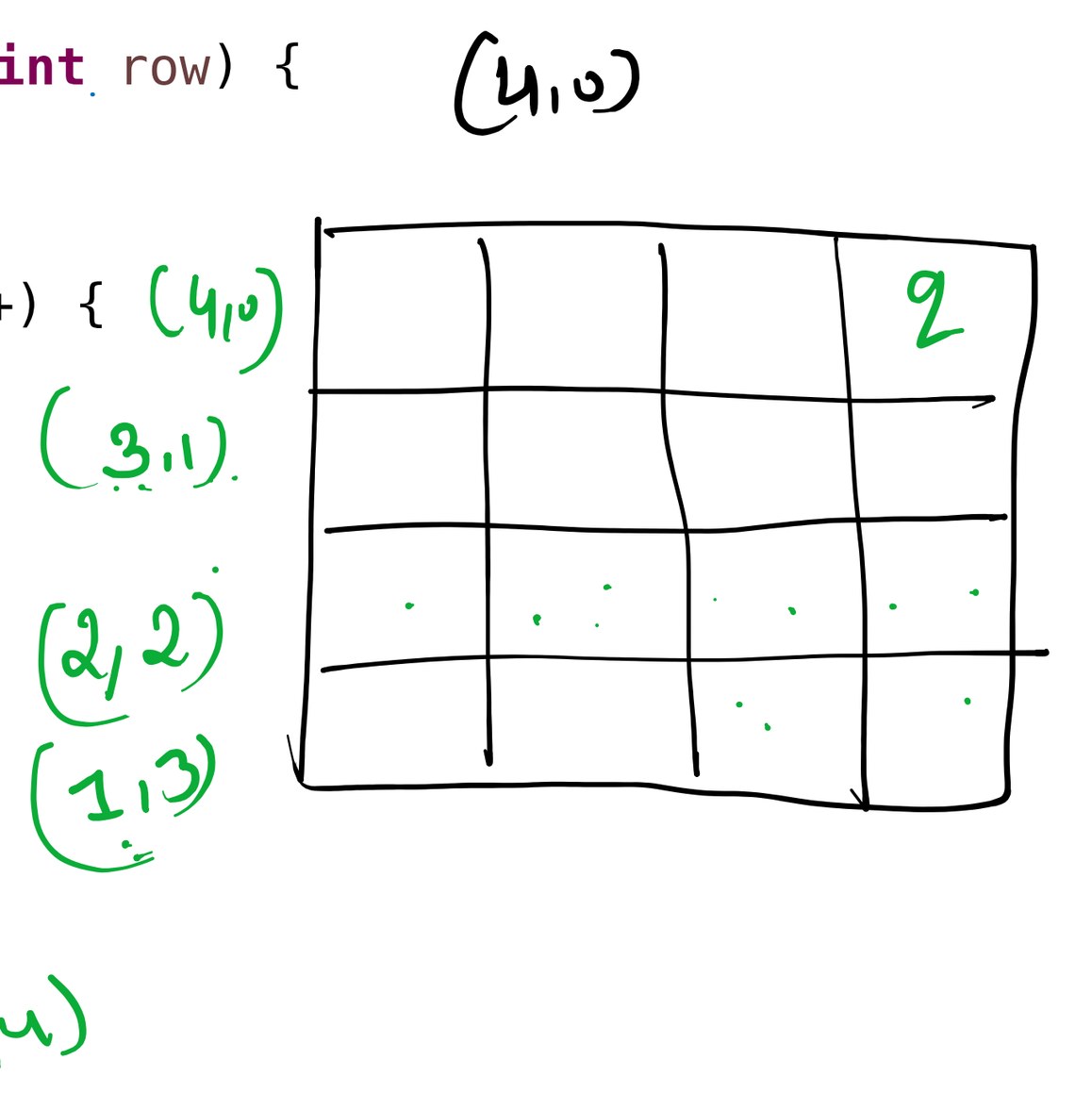




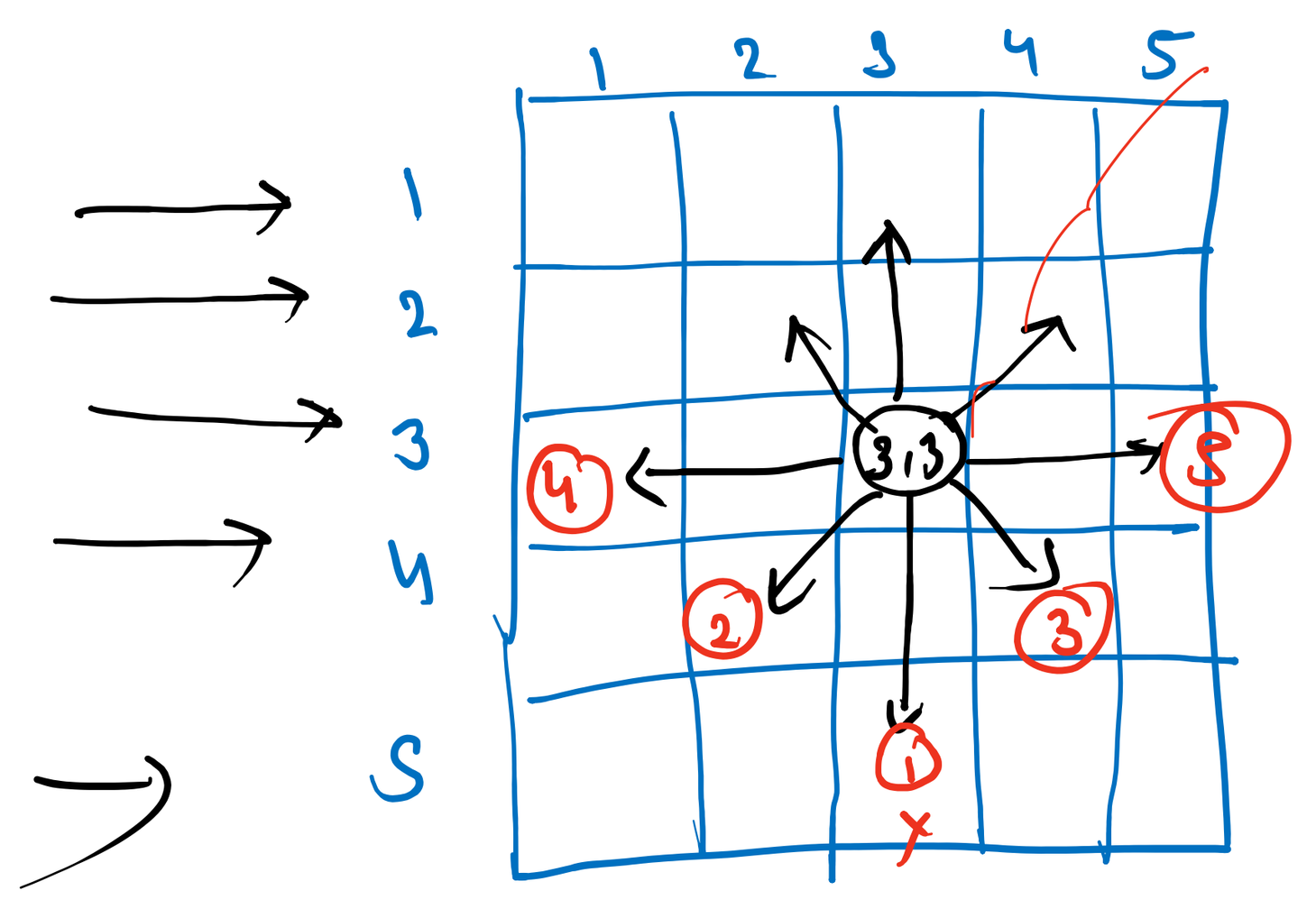


```java
public static void Print(boolean[][] board, int tq, int row) {
        if (tq == 0) {
            Display(board);
        }
        for (int col = 0; col < board[0].length; col++) {

            if (issafe(board, row, col)) {
                board[row][col] = true;
                Print(board, tq - 1, row + 1);
                board[row][col] = false;
            }
        }
    }
```




```java
public static boolean issafe(boolean[][] board, int row, int col) {
        // TODO Auto-generated method stub
        return false;
    }
```



```java
public static void Print(int[][] grid, int row, int col) {
    if(grid[row][col]!=0) {
        Print(grid, row, col+1);
    }
    else {
        for (int val = 1; val <=9; val++) {
            if(issafe(grid,row,col,val)) {
                grid[row][col]=val;
                Print(grid, row, col+1);
                grid[row][col]=0;
            }
        }
    }
}
```

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

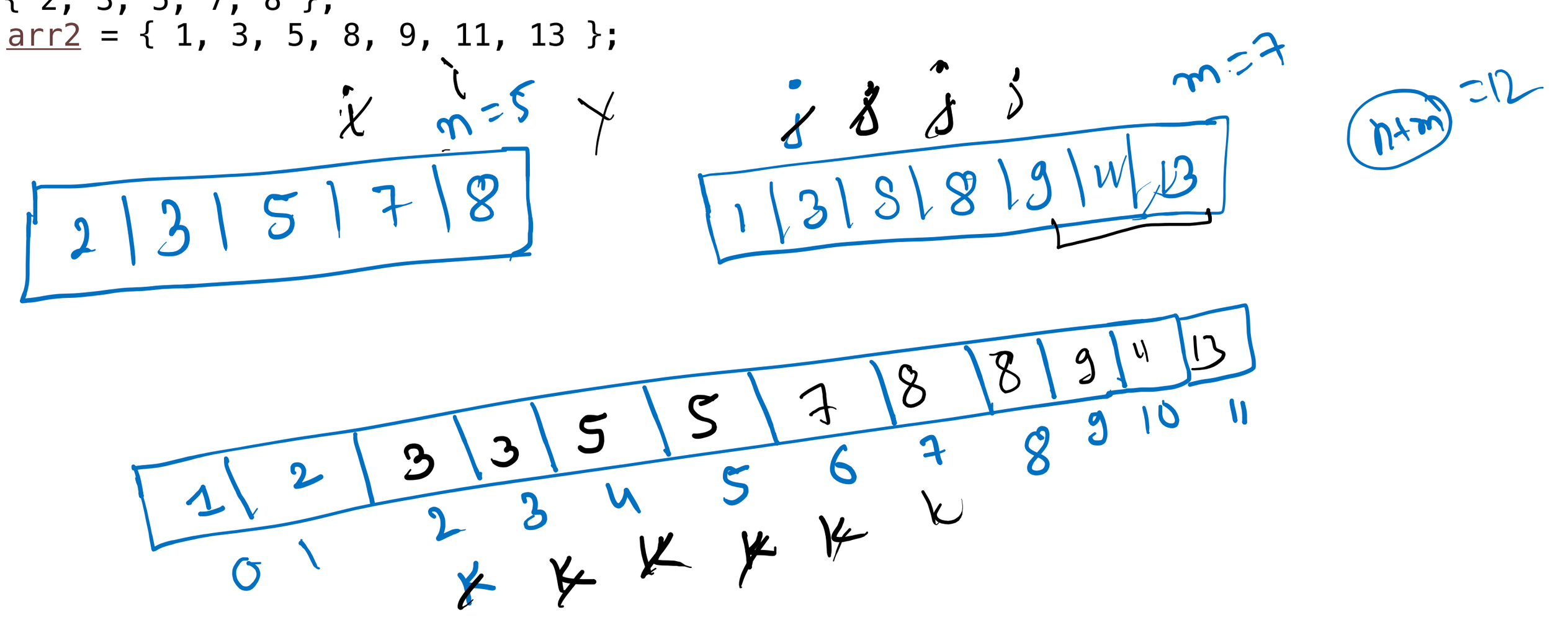
col: [0-1-2] → 0 [3 4 5] → 3 [6,7,8] → 6

Row = [0 1 2] → 0 [3 4 5] → 3 [6 7 8] → 6

Row = Row - Row % 3

Col = Col - Col % 3

```java
int[] arr1 = { 2, 3, 5, 7, 8 };
    int[] arr2 = { 1, 3, 5, 8, 9, 11, 13 };
```



```java
public static int[] Merge(int[] arr1, int[] arr2) {
    int n = arr1.length;
    int m = arr2.length;
    int[] ans = new int[n + m];
    int i = 0, j = 0, k = 0;
    while (i < n && j < m) {
        if (arr1[i] < arr2[j]) {
            ans[k] = arr1[i];
            k++;
            i++;
        } else {
            ans[k] = arr2[j];
            k++;
            j++;
        }
    }
}
```