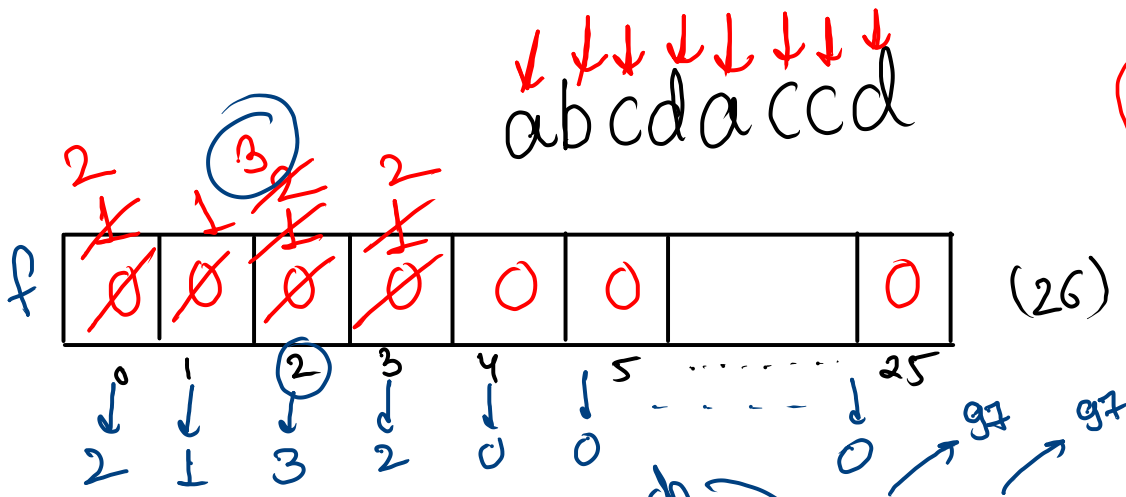
Maximum Freq Character



arr[i] = arr[i] + 1;

arr[i]++;

a → 0
b → 1
c → 2
d → 3
e → 4
⋮
z → 25

ch

'a' − 'a' = 0
'b' − 'a' = 1
'c' − 'a' = 2
'd' − 'a' = 3
'a' − 'a' = 0
'c' − 'a' = 2
'c' − 'a' = 2
'd' − 'a' = 3

97 97

98

ascii values

indexes

5 → f
2 → c
3 → d
1 → b
0 → a

```java
public static void maxFreq(String str) {
    int[] freq = new int[26];   // because we have only 26 characters to count
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int idx = ch - 'a';

        freq[idx] = freq[idx] + 1;
        // freq[idx]++;
    }

    int maxFreq = -1;
    int idx = -1;

    for (int i = 0; i < 26; i++) {
        int f = freq[i];
        if ( f > maxFreq ) {
            maxFreq = f;
            idx = i;
        }
    }

    char ans = (char)(idx + 'a');
    System.out.println(ans);
}
```

n

26

$T.C = O(n+26)$

$\simeq O(n)$

where, n is length of str

$S.C = O(26)$

$\simeq O(1)$

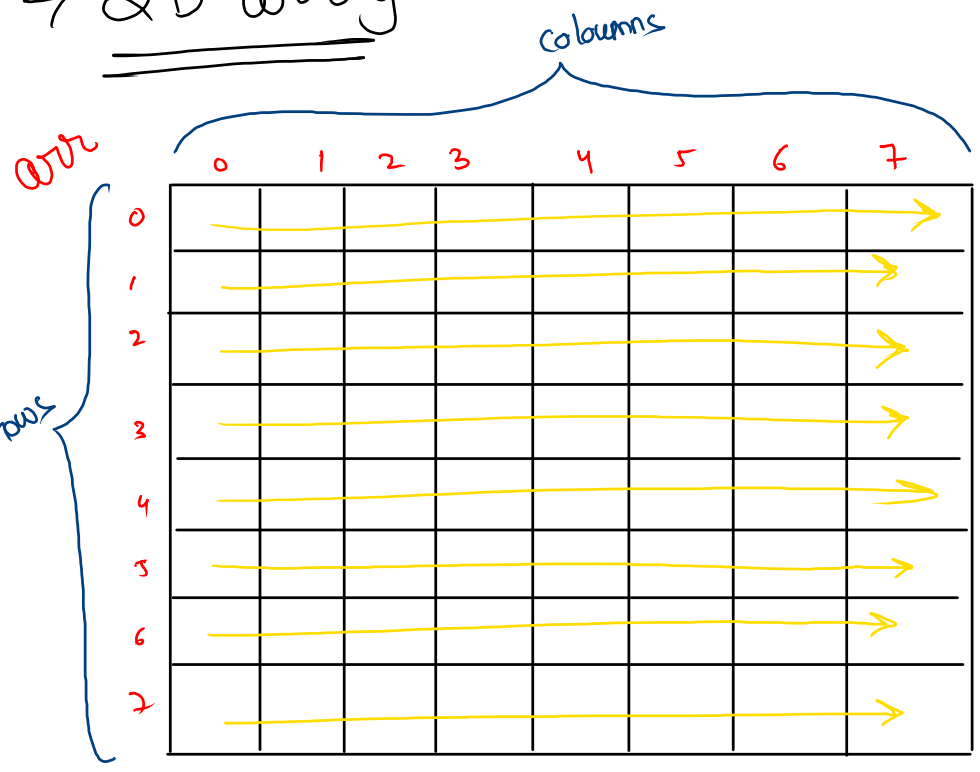
2 + 97 = 99 = 'c'

'a' = 97

+2

99 = 'c'

⇒ 2D array



declare:-

datatype[][] arr_name = new datatype[rows][col];

int[][] arr = new int[n][n];

no. of rows = arr.length ;

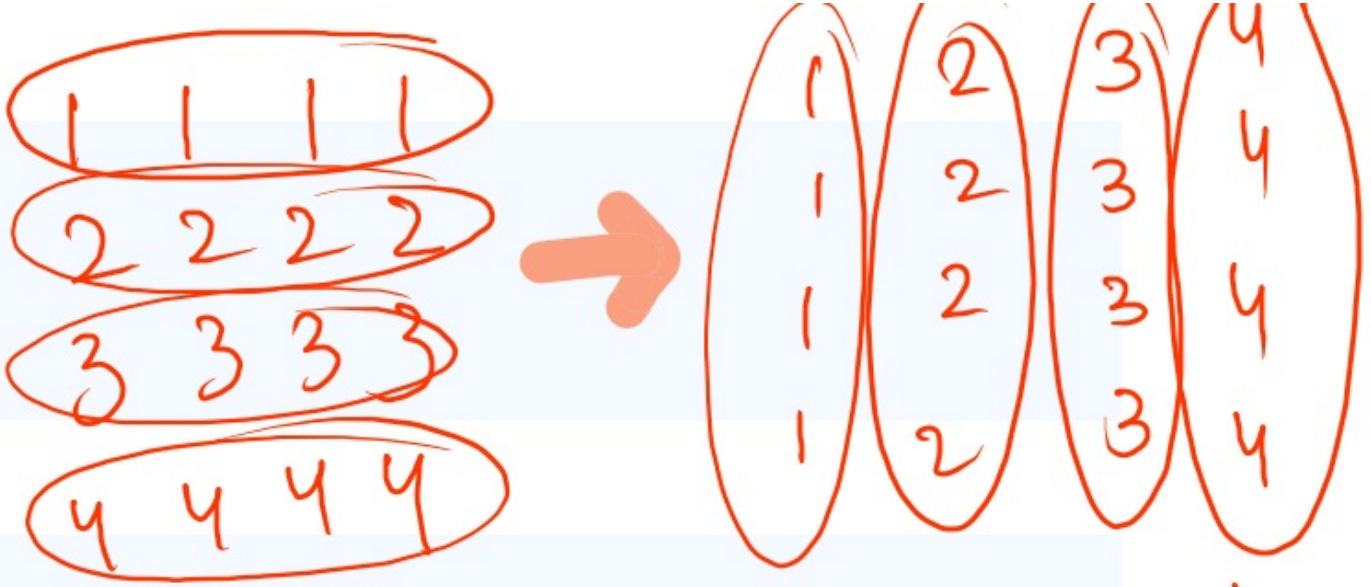
no. of col = arr[0].length ;

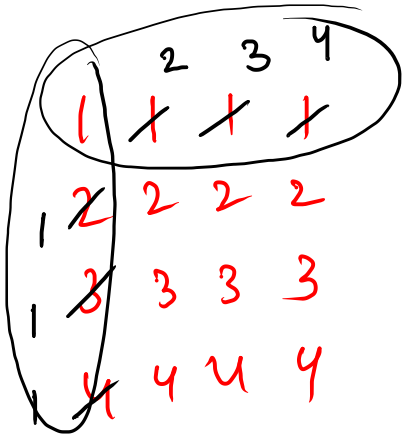
```
for( int i=0 ; i < arr.length ; i++) {
    for( int j=0 ; j< arr[0].length; j++) {
        Syso( arr[i][j] );
    }
}
```

rows

col

Transpose of Matrix of N*N





Convert all rows into coloumn

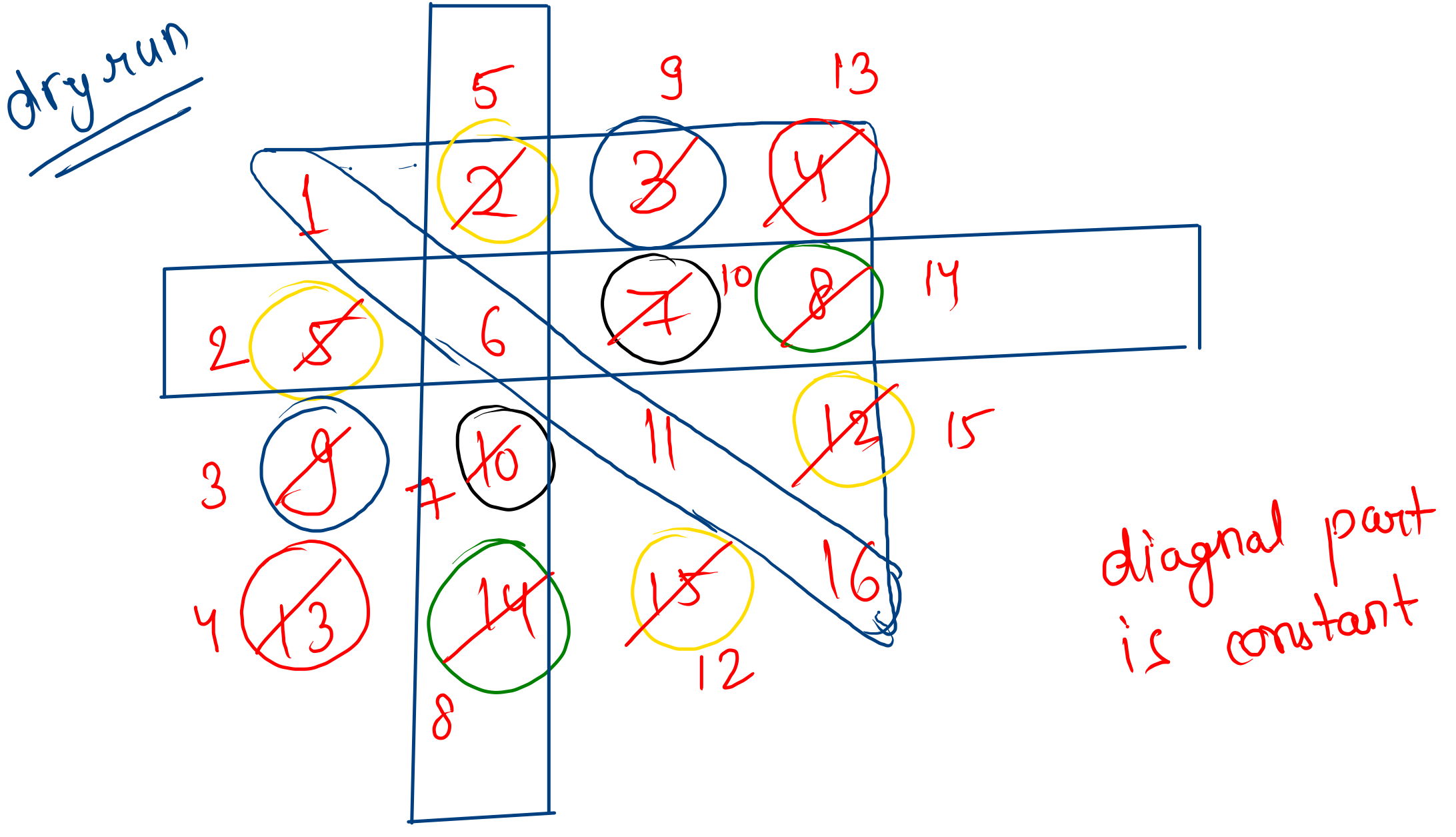
or all colouns into rows

```java
public static void transpose(int[][] arr, int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i; j++) {

            int temp = arr[i][j];
            arr[i][j] = arr[j][i];
            arr[j][i] = temp;

        }
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
}
```


dry run
5
9
13
1
2
3
4
10
14
2
8
6
7
8
11
12
15
3
9
7
10
4
13
14
15
16
12
8
diagnal part is constant