

Modify The Matrix

~~(V.9mp)~~

Q501

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

modify

(3,1)

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

(6,8)

(0,5)

approach 1

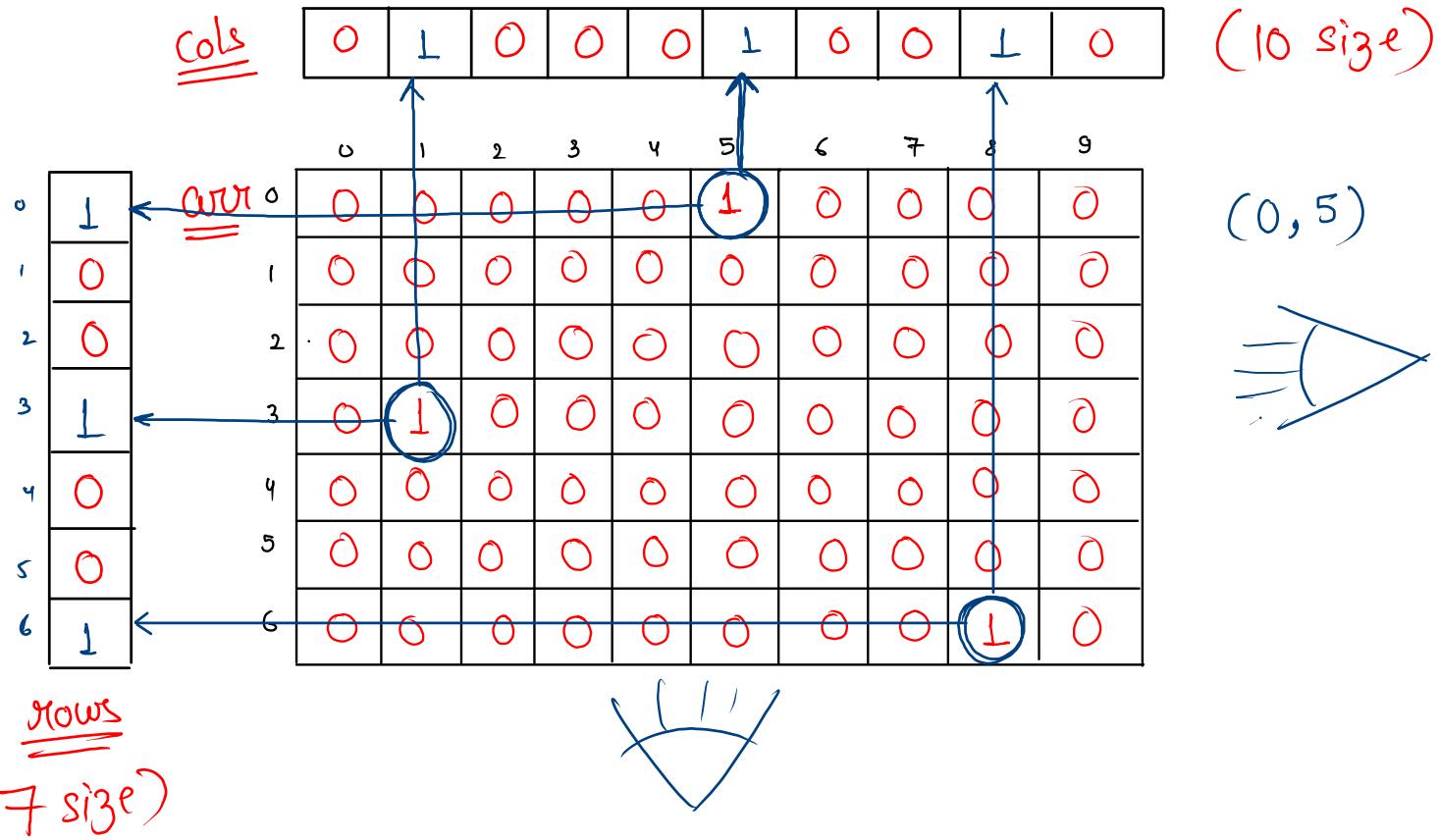
- Create another matrix of same size
- Traverse on original array
- And modify new array acc. to cond'

$$T.C = O(m * n)$$

$$S.C = O(m * n)$$

~~approach 2~~

~~step 1~~



1d
cols

0	1	0	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---

(10 size)

1d

0	1
1	0
2	0
3	1
4	0
5	0
6	1

arr

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

(0, 5)

array got
modified

rows
(7 size)

$$S.C = O(m+n)$$

pseudo
code

- 1) create 1d array of size m
- 2) " " " " n
- 3) traverse in 2d array
 - 3.1) if arr[i][j] == 1
then update 1d arrays
 $\text{rows}[i] = 1;$
 $\text{cols}[j] = 1;$
- 4) traverse in 2d array
 - 4.1) if rows[i] == 1 OR cols[j] == 1
then update 2d array
 $\text{arr}[i][j] = 1;$

Code

```
public static void modifyTheMatrix(int[][] arr, int m, int n) {  
    int[] rows = new int[m];  
    int[] cols = new int[n];  
    for (int i = 0; i < m; i++) {  
        for (int j = 0; j < n; j++) {  
            if (arr[i][j] == 1) {  
                rows[i] = 1;  
                cols[j] = 1;  
            }  
        }  
    }  
    for (int i = 0; i < m; i++) {  
        for (int j = 0; j < n; j++) {  
            if (rows[i] == 1 || cols[j] == 1) {  
                arr[i][j] = 1;  
            }  
        }  
    }  
}
```

$$T.C = O(m \times n)$$

$$S.C = O(m + n)$$

⇒ String
= 0

str = "abcABC";
 0 1 2 3 4 5

inbuilt
function

- 1) str.length() // 6
- 2) str.charAt(index) // 'B'
 ↓
 4
- 3) str.toLowerCase() // "abcabc"
- 4) str.toUpperCase() // "ABCABC"

$\text{str} = \text{"abcABC"};$

0 1 2 3 4 5

substring :-

Sub part of string

- sub part will always be continuous
- always moves in forward direction

str = "abcfABC";

0 1 2 3 4 5

(in Java)

syntax

str.substring(start index, end index + 1);

Cx:-

str.substring(1, 5); // "bcfA"

str.substring(0, 4); // "abcf"

str.substring(2, 4); // "cf"

str.substring(3, 4); // "f"

str.substring(1, 1); // "" (empty string)

str.substring(1, 5); // bcfa

str.substring(1, 6); // bcfaBC

str.substring(1, 7); // error

syntax 2

str.substring(start index);

str = "abcABC";
 ^
 0 1 2 3 4 5

str.substring(3); // "ABC"

str.substring(0); // "abcABC";

Code

$$\underline{T.C = O(\text{str.length}())}$$

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    printChar(str);
}

public static void printChar(String str) {
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        System.out.println(ch);
    }
}
```