Good String Checker

freq =

0	1	2	3	4		25
~~0~~	~~0~~	~~0~~	~~0~~	0	- - - - - - - - - - - - - - - -	0
~~1~~ 2	~~1~~ 2	~~1~~ 2	~~1~~ 2			

str = "abcdcbad" ,

0 1 2 3 4 5 6 7

→

```
ch = str.charAt(i);
idx = ch - 'a' ;
```

dry run

i = 0, ch = 'a'
idx = 0

i = 1, ch = 'b'
idx = 1

i = 2, ch = 'c'
idx = 2

i = 3, ch = 'd'
idx = 3

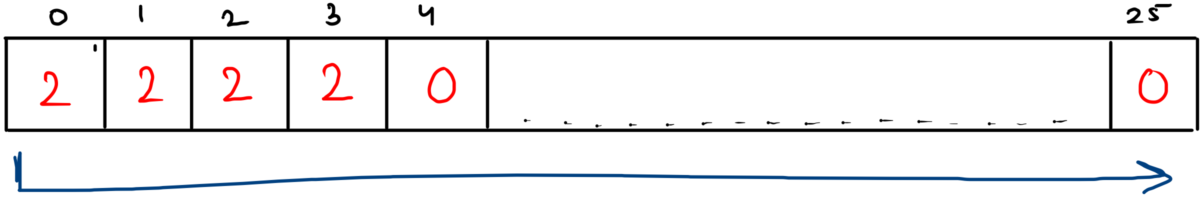
i = 4, ch = 'c'
idx = 2

i = 5 , ch = 'b'
idx = 1

i = 6, ch = 'a'
idx = 0

i = 7, ch = 'd'
idx = 3

freq =

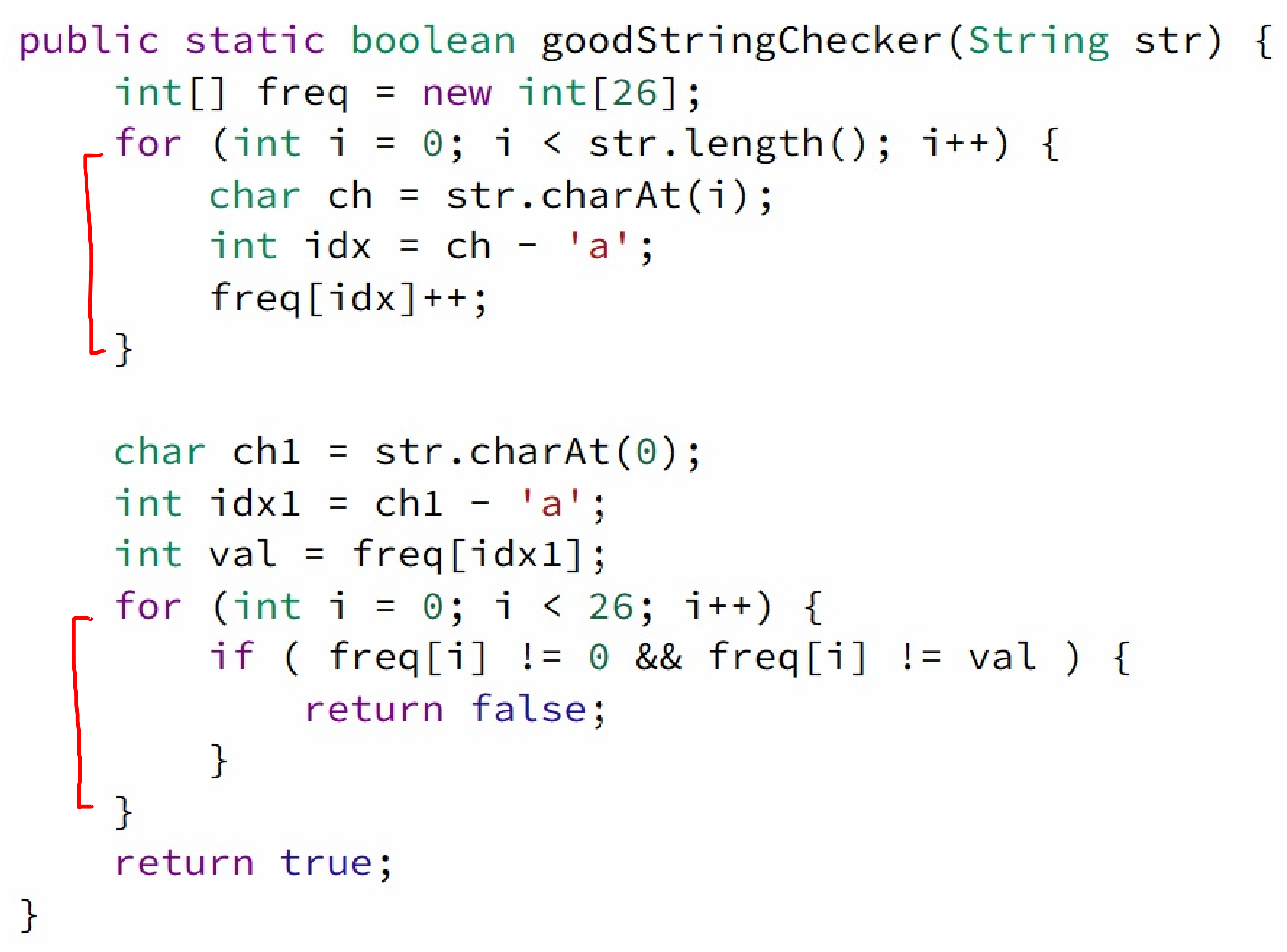


0	1	2	3	4	. . .	25
2	2	2	2	0	. . .	0

```
ch = str.charAt(0);          }
val = freq[ch - 'a'];        }  constant time          str = "abcdcbad"

for (int i = 0; i < freq.length; i++){
        if ( freq[i] != val ) {
            return false;
        }
}
retue true;
```

travasing on freq array to check

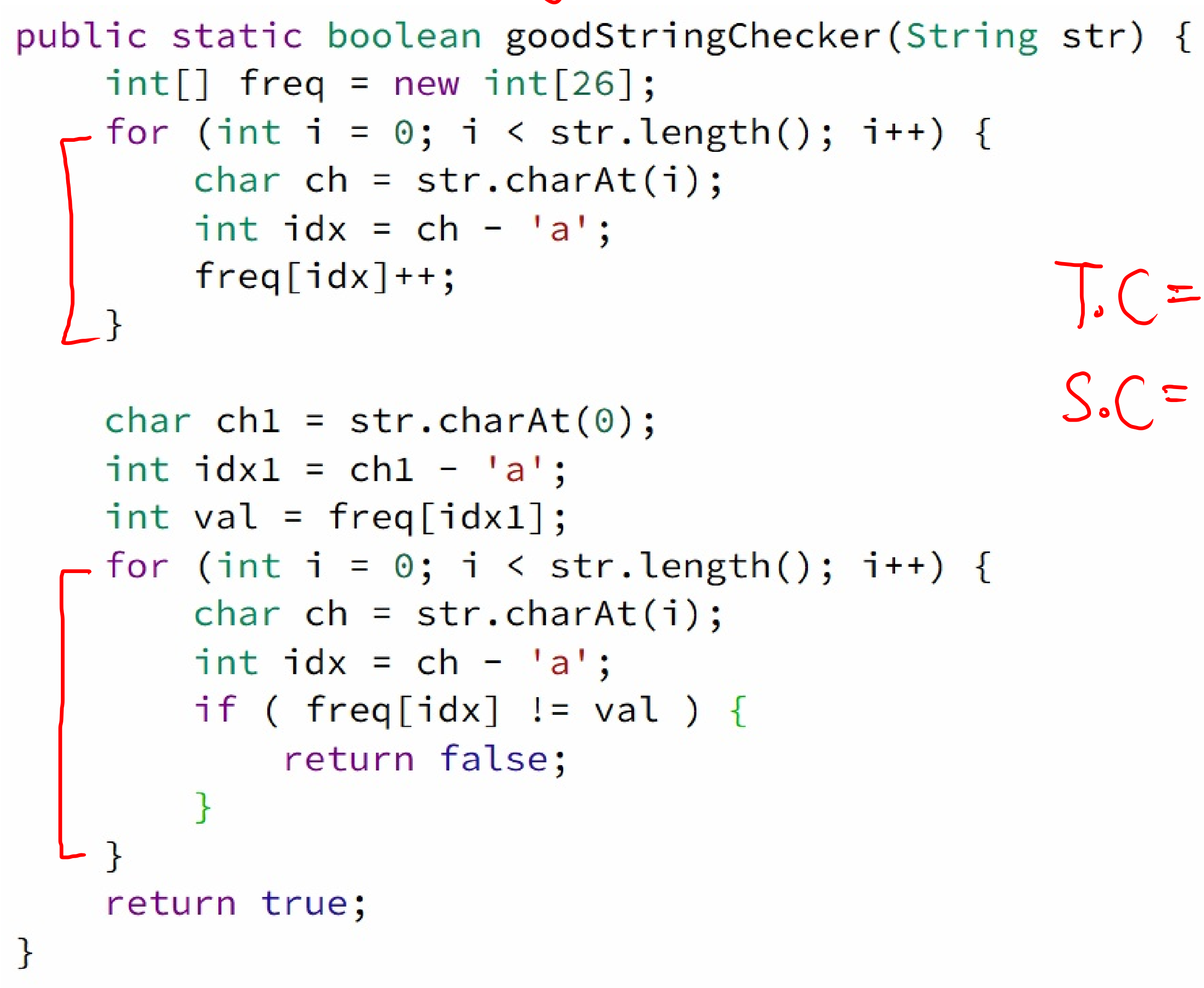


```java
public static boolean goodStringChecker(String str) {
    int[] freq = new int[26];
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int idx = ch - 'a';
        freq[idx]++;
    }

    char ch1 = str.charAt(0);
    int idx1 = ch1 - 'a';
    int val = freq[idx1];
    for (int i = 0; i < 26; i++) {
        if ( freq[i] != 0 && freq[i] != val ) {
            return false;
        }
    }
    return true;
}
```

traversing on string to check

code



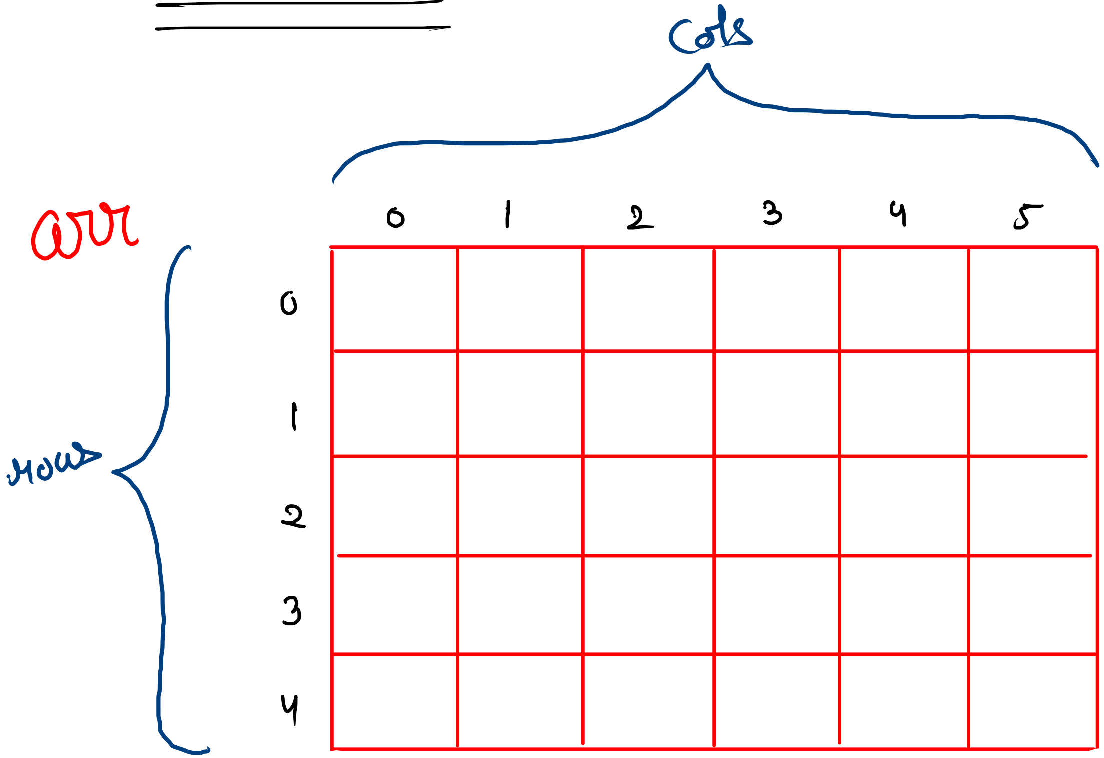
```
public static boolean goodStringChecker(String str) {
    int[] freq = new int[26];
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int idx = ch - 'a';
        freq[idx]++;
    }

    char ch1 = str.charAt(0);
    int idx1 = ch1 - 'a';
    int val = freq[idx1];
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int idx = ch - 'a';
        if ( freq[idx] != val ) {
            return false;
        }
    }
    return true;
}
```

T.C = O(n)

S.C = O(1)

⇒ 2D array



rows = 5
cols = 6

size = 5 x 6

Note:-

no. of rows = arr.length; // 5

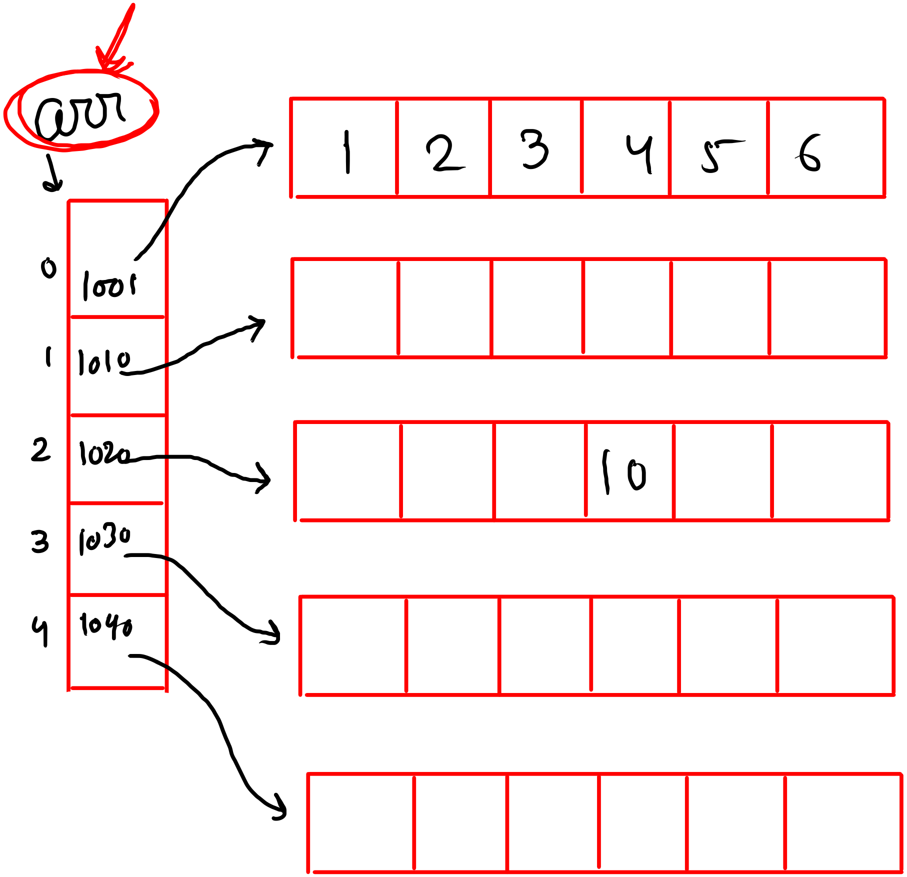
no. of cols = arr[0].length; // 6

fun fact

arr	0	1	2	3	4	5
0	1	2	3	4	5	6
1						
2				10		
3						
4						

no. of boxes = 5×6

= 30



⇒ how to access an index in 2d array

arr =

	0	1	2	3	4	5
0	1	2	3	4	5	6
1	7	8	9	10	11	12
2	13	14	15	16	17	18
3	19	20	21	22	23	24
4	25	26	27	28	29	30

access

```
int x = arr[2][3]
```

row index

col index

declaration

1D :- `int[] arr = new int[5];`

2D :- `int[][] arr = new int[5][6];`

no. of rows

no. of cols

update value of 16 to 32

```
arr[2][3] = 32
```

⇒ Traverse in 2D array

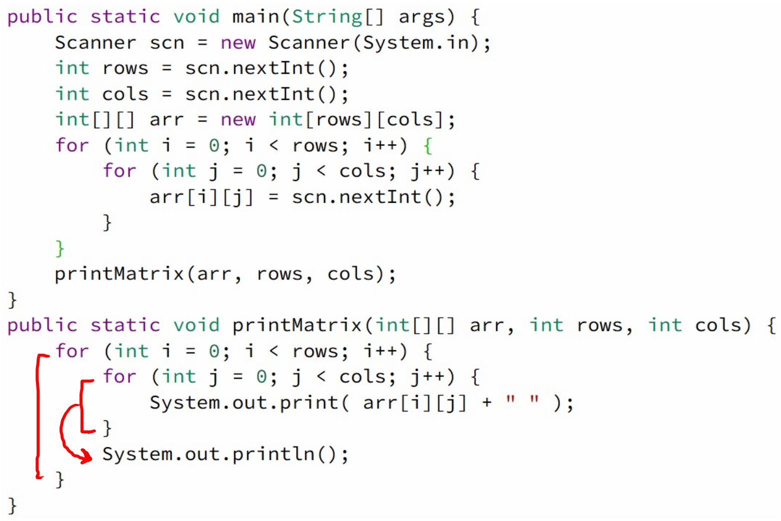
```
int rows = arr.length;
int cols = arr[0].length;

for( int i=0; i< rows ; i++){      ← rows
    for(int j=0; j< cols ; j++){      ← cols
        Syso ( arr[i][j] );
    }
    Sysoln();
}
```

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9
3	10	11	12

size = 4 * 3

Code



```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int rows = scn.nextInt();
    int cols = scn.nextInt();
    int[][] arr = new int[rows][cols];
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            arr[i][j] = scn.nextInt();
        }
    }
    printMatrix(arr, rows, cols);
}
public static void printMatrix(int[][] arr, int rows, int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            System.out.print( arr[i][j] + " " );
        }
        System.out.println();
    }
}
```

arr

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9
3	10	11	12

i=0,	j=0,	(0,0)
	j=1,	(0,1)
	j=2,	(0,2)
i=1,	j=0,	(1,0)
	j=1,	(1,1)
	j=2,	(1,2)
i=2,	j=0,	(2,0)
	j=1,	(2,1)
	j=2,	(2,2)
i=3,	j=0,	(3,0)
	j=1,	(3,1)
	j=2,	(3,2)

O/P

```
1 _ 2 _ 3 _
4 _ 5 _ 6 _
7 _ 8 _ 9 _
10 _ 11 _ 12 _
```

T.C :- O(rows * cols)

:- O(m * n)

linear

linear :- we are visiting every element once

quad :- we are visiting every element for every element