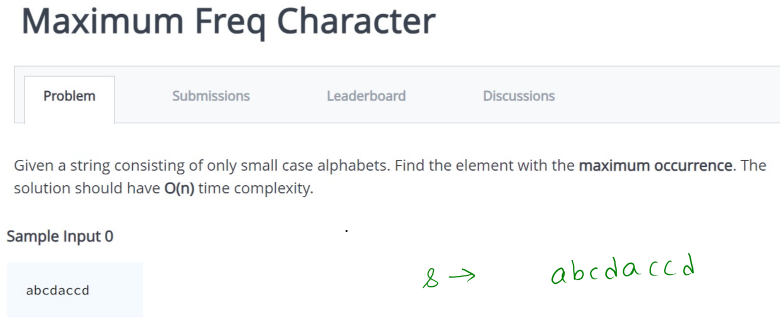


Maximum Freq Character

Given a string consisting of only small case alphabets. Find the element with the **maximum occurrence**. The solution should have **O(n)** time complexity.

Sample Input 0

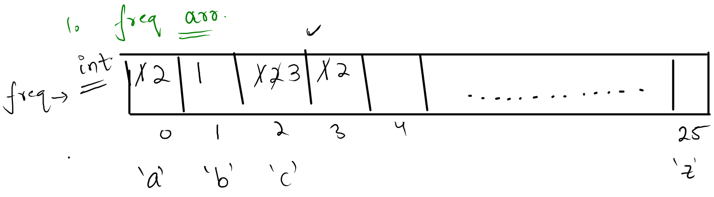
```
abcdaccd
```

8 → abcdaccd

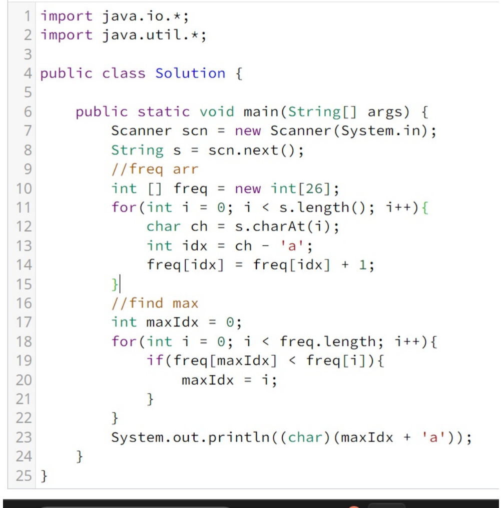
Sample Output 0

```
c
```

1. freq arr.



2. Take max from freq array.



```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        String s = scn.next();
        //freq arr
        int [] freq = new int[26];
        for(int i = 0; i < s.length(); i++){
            char ch = s.charAt(i);
            int idx = ch - 'a';
            freq[idx] = freq[idx] + 1;
        }
        //find max
        int maxIdx = 0;
        for(int i = 0; i < freq.length; i++){
            if(freq[maxIdx] < freq[i]){
                maxIdx = i;
            }
        }
        System.out.println((char)(maxIdx + 'a'));
    }
}
```

8 → abcdaccd

Good String Checker

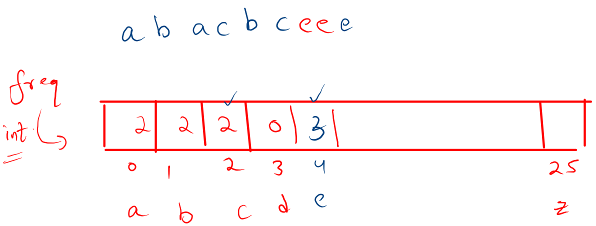
Problem | Submissions | Leaderboard | Discussions

Given a string `str`, return **true** if `str` is a **good string**, or **false** otherwise.

A string str is good if all the characters that appear in `str` have the same number of occurrences (i.e., the same frequency).

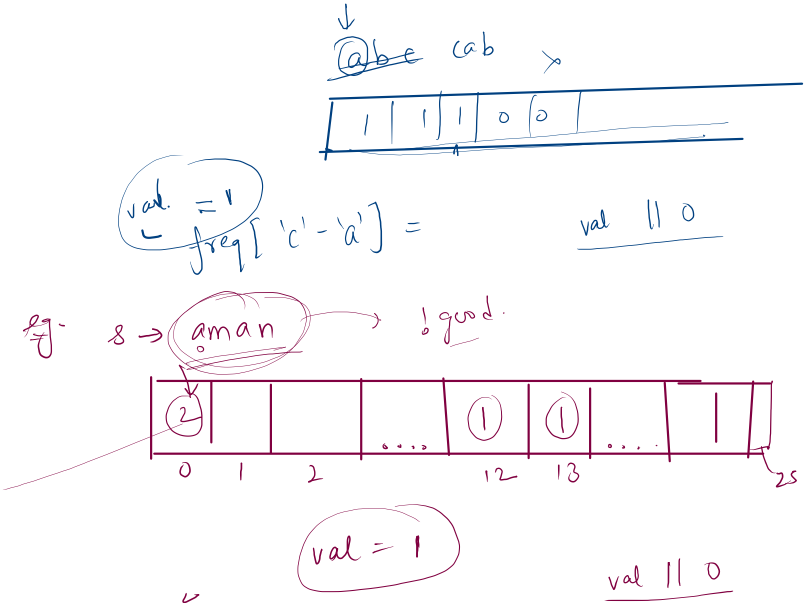
Sample Input 0

```
abacbc
```



Sample Output 0

```
true
```



eg. 1.

s→ d e b c c b

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

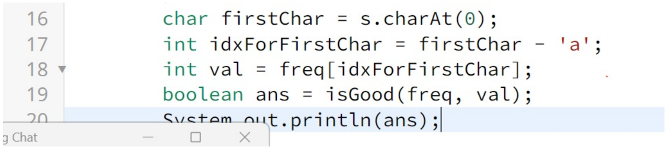
val = 1

eg2.

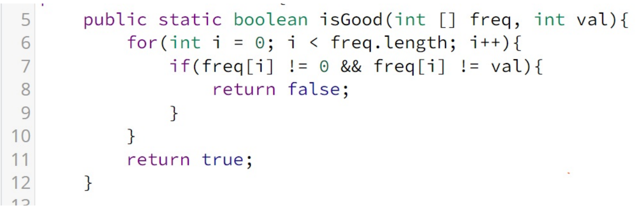
s→ d e b c c b d e

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

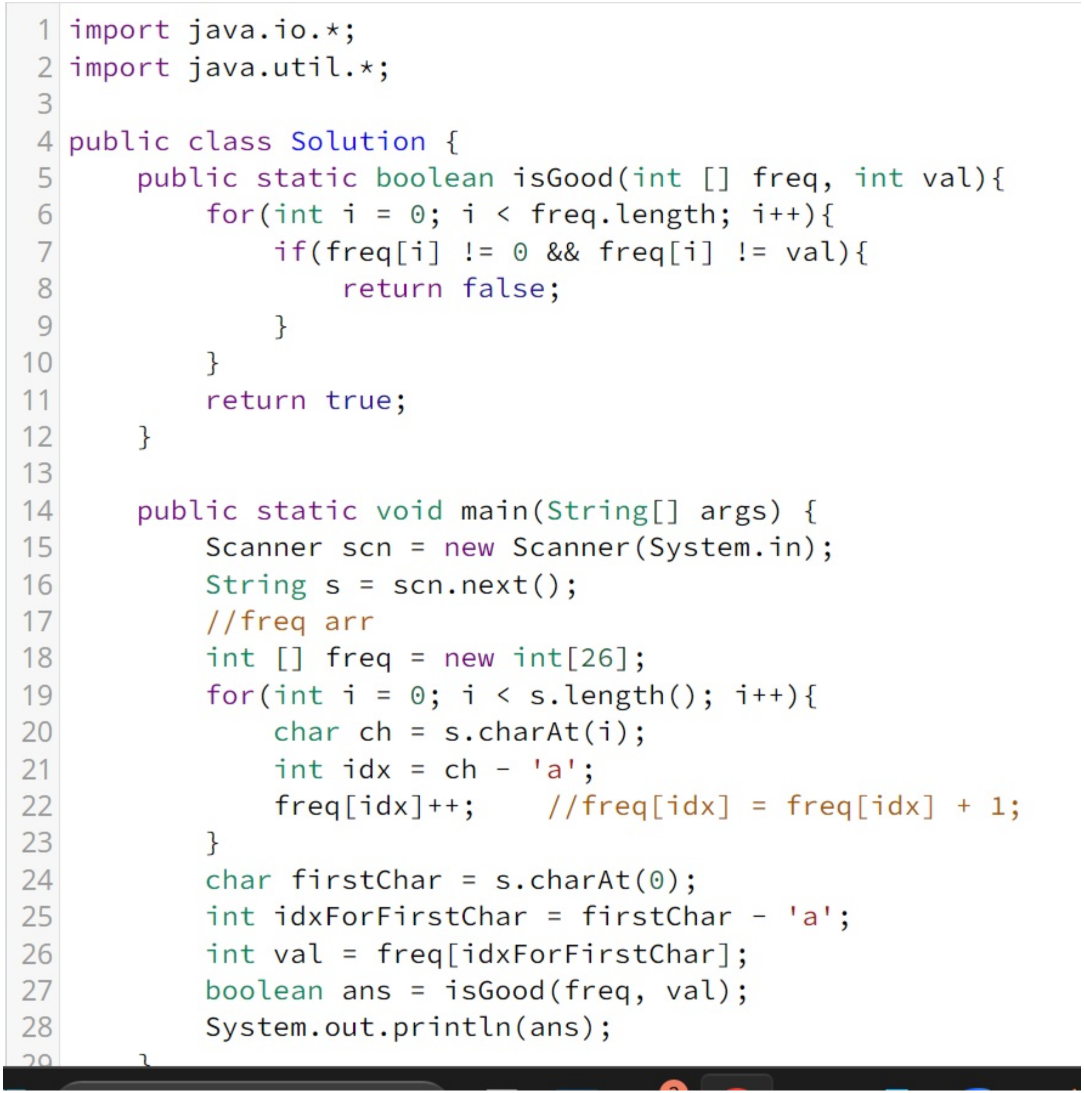
val = 2



```
        char firstChar = s.charAt(0);
        int idxForFirstChar = firstChar - 'a';
        int val = freq[idxForFirstChar];
        boolean ans = isGood(freq, val);
        System.out.println(ans);
```

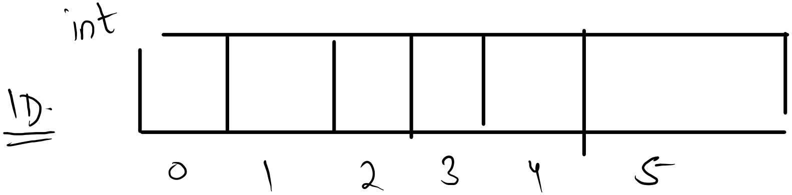


```
    public static boolean isGood(int [] freq, int val){
        for(int i = 0; i < freq.length; i++){
            if(freq[i] != 0 && freq[i] != val){
                return false;
            }
        }
        return true;
    }
```

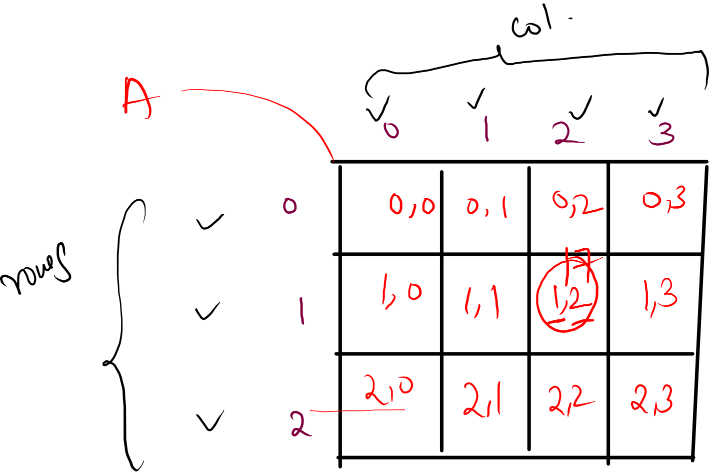


```java
import java.io.*;
import java.util.*;

public class Solution {
    public static boolean isGood(int [] freq, int val){
        for(int i = 0; i < freq.length; i++){
            if(freq[i] != 0 && freq[i] != val){
                return false;
            }
        }
        return true;
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        String s = scn.next();
        //freq arr
        int [] freq = new int[26];
        for(int i = 0; i < s.length(); i++){
            char ch = s.charAt(i);
            int idx = ch - 'a';
            freq[idx]++;    //freq[idx] = freq[idx] + 1;
        }
        char firstChar = s.charAt(0);
        int idxForFirstChar = firstChar - 'a';
        int val = freq[idxForFirstChar];
        boolean ans = isGood(freq, val);
        System.out.println(ans);
    }
```

2D - Array / Matrix.



int [ ] A = new int [6];



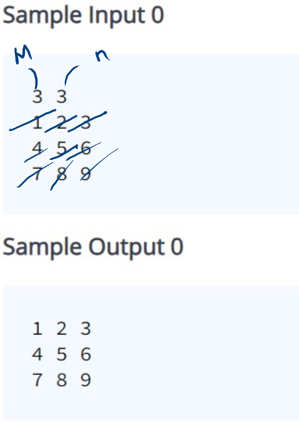
	0	1	2	3
0	0,0	0,1	0,2	0,3
1	1,0	1,1	17 1,2	1,3
2	2,0	2,1	2,2	2,3

A[1][2]=17

r x c

3x4

Print Matrix



Sample Input 0

M n

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

Sample Output 0

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

1. { m → rows
n → cols.

2. { 2D Matrix

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

3X3

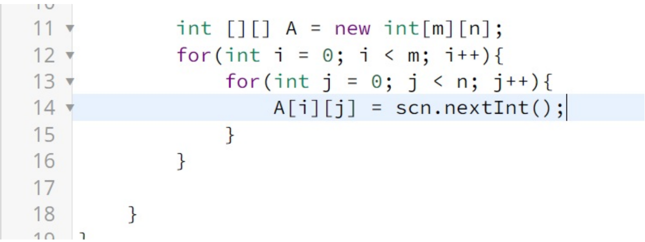
3. } i/p

4. { o/p → print

i j

{ 0,0 → 0,1 → 0,2 , 1,0 → 1,1 → 1,2
2,0 → 2,1 → 2,2 }

i	j
0	0 1 2
1	0 1 2
2	0 1 2



```java
        int [][] A = new int[m][n];
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                A[i][j] = scn.nextInt();
            }
        }

    }
```

$m = 3$

$n = 3$

$$\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

$$\begin{array}{c|c|c|c} 0 & \boxed{1} & 2 & 3 \\ \hline 1 & 4 & 5 & 6 \\ \hline 2 & 7 & 8 & 9 \\ \hline & 0 & 1 & 2 \end{array}$$

$i = \cancel{0}$

1 2

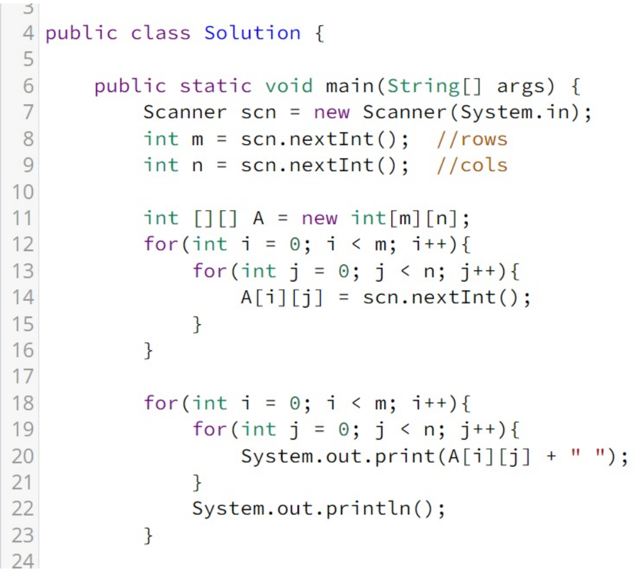
$0 < 3$ ✓

$1 < 3$ ✓

$j = 0$

$= 1$

$= 2$



```java
public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int m = scn.nextInt();  //rows
        int n = scn.nextInt();  //cols

        int [][] A = new int[m][n];
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                A[i][j] = scn.nextInt();
            }
        }

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

1 2 3

4 5 6

7 8 9

$n \times m$

TC → $O(n \times m)$

✓ SC → $O(1)$ → not include i/p

SC → $O(n^2)$ → include i/p

Print Alternate Row

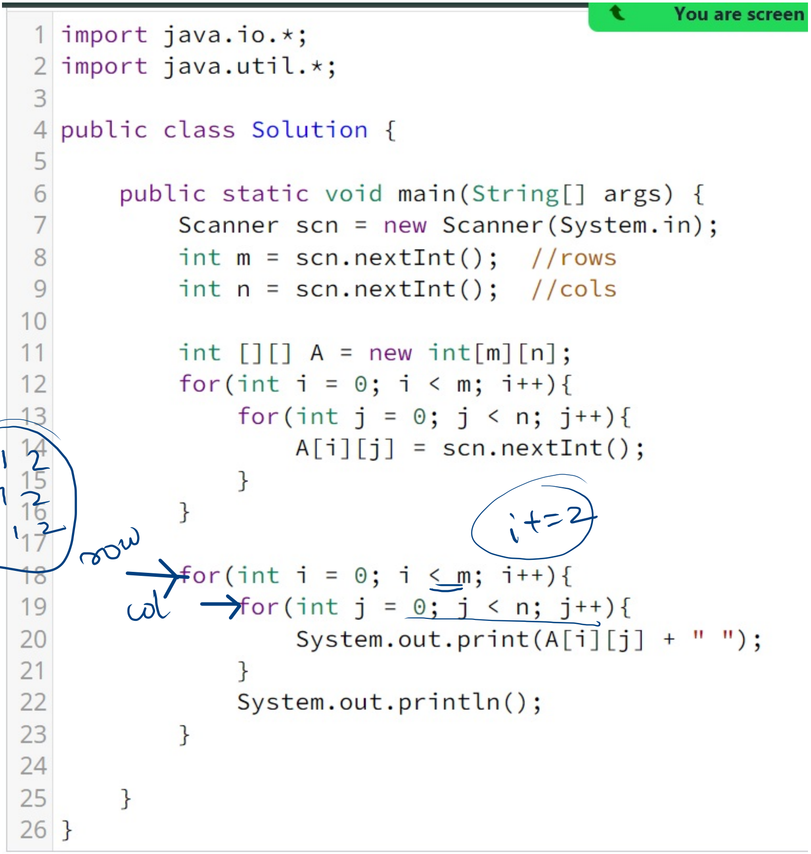
m=4
n=6

Sample Input 0

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

Sample Output 0

```
2 3 8 7 0 4
0 0 8 1 0 8
```



```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int m = scn.nextInt();  //rows
        int n = scn.nextInt();  //cols

        int [][] A = new int[m][n];
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                A[i][j] = scn.nextInt();
            }
        }

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

    }
}
```

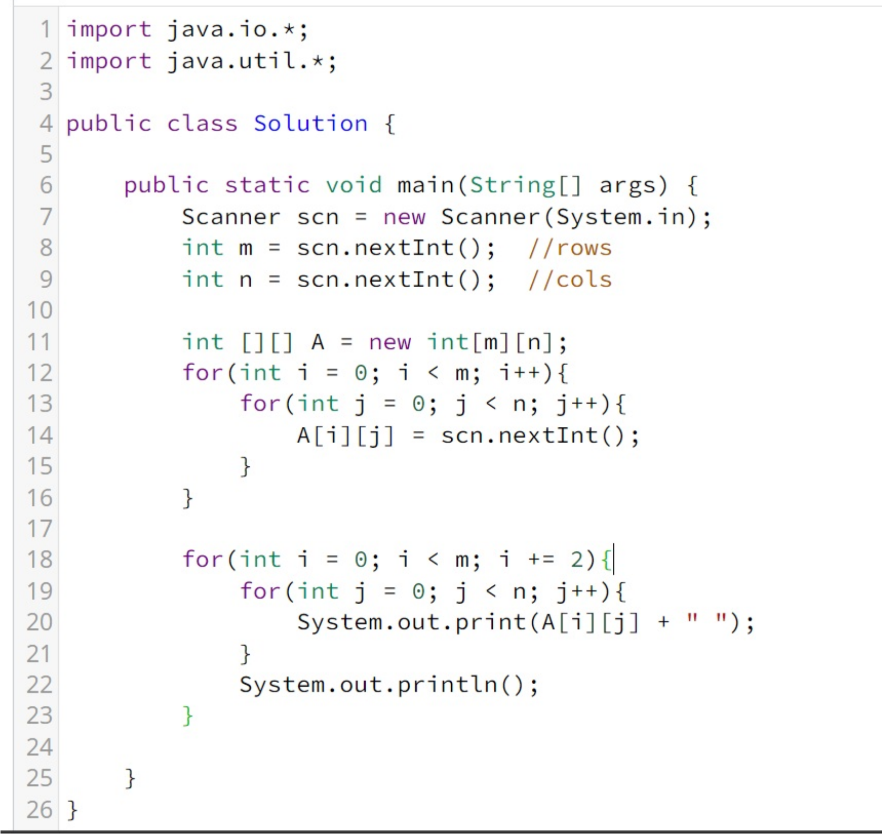
0 0 1 2
1 0 1 2
2 0 1 2

row

col

i+=2

0 → 0 1 2
x 1
2 → 0 1 2



```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int m = scn.nextInt();  //rows
        int n = scn.nextInt();  //cols

        int [][] A = new int[m][n];
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                A[i][j] = scn.nextInt();
            }
        }

        for(int i = 0; i < m; i += 2){
            for(int j = 0; j < n; j++){
                System.out.print(A[i][j] + " ");
            }
            System.out.println();
        }

    }
}
```

1 2
5 6

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & 2 \\ 1 & 3 & 4 \\ 2 & 5 & 6 \\ 3 & 7 & 8 \end{array}$$

Print the matrix left-diagonal wise

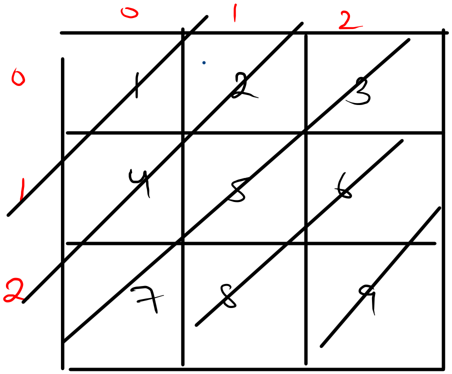
Sample Input 0

nxn.

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

Sample Output 0

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



3X3.
nxn.

$n \times n.$ n^2

$2 \times n - 1 \rightarrow$ rows

$3 \times 3 \rightarrow 5$

$2 \times 2 \rightarrow 3$

$$\underbrace{0,0}_{0} \rightarrow \underbrace{0,1 \rightarrow 1,0}_{1} \rightarrow \underbrace{0,2 \rightarrow 1,1 \rightarrow 2,0}_{2} \rightarrow \underbrace{1,2 \rightarrow 2,1}_{3} \rightarrow \underbrace{2,2}_{4}$$

$$\overset{0}{i} + \overset{0}{j} =$$