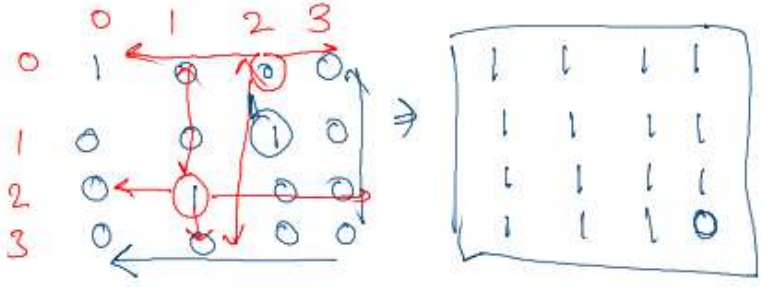
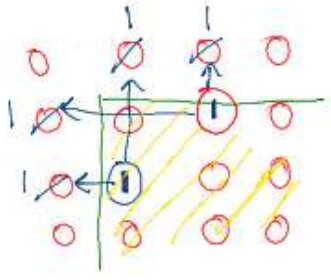
Q= if anywhere in the matrix is 1.
Then make that whole row and column as 1.





Step ① → check if there is any 1 in the 1st row

Step ② → check if there is any 1 in the 1st col

arr[0][i] arr[i][0]

Step ③ modify row 0th and col 0th with 1 if arr[i][j] = 1

④ if (arr[i][0] == 1 || arr[0][j] == 1) then modify arr[i][j] = 1

✓ ⑤ modify 0th row and 0th col

~~if~~ arr[0][j] = 1

and

if (i ~~and~~ col) arr[i][0] = 1

```
for(int j=0;j<col;j++){
    if(arr[0][j]==1){
      firstrowFlag=true;
        break;
    }
}
```

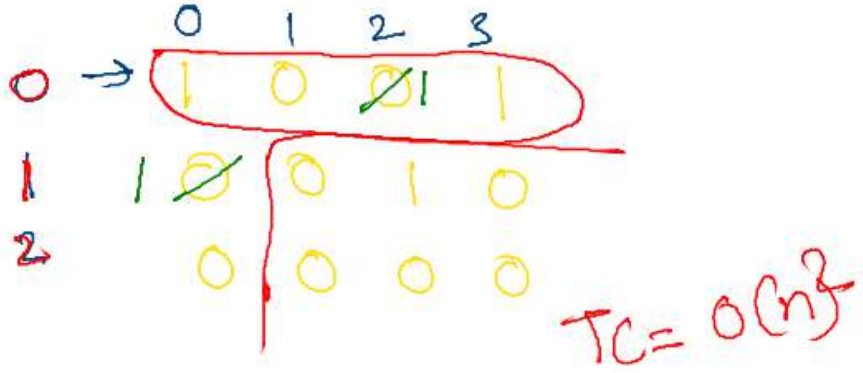
O(n)

firstrowFlag = true;

```
for(int i=0;i<row;i++){
    if(arr[i][0]==1){
      firstcolFlag=true;
        break;
    }
}
```

O(n)

colFlag = true

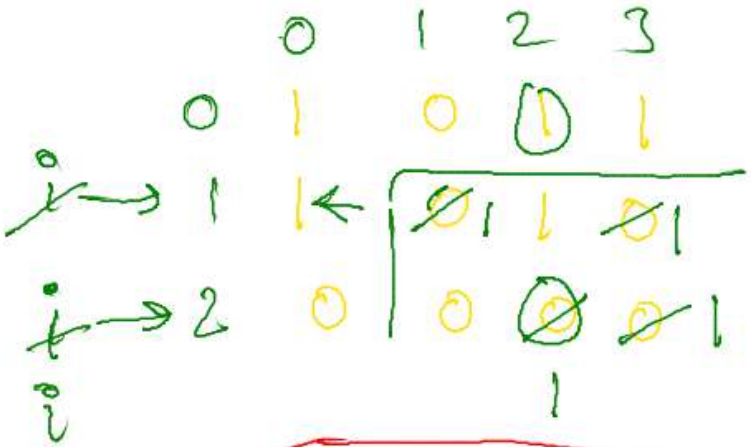



```
for(int i=1;i<arr.length;i++){
    for(int j =1;j<arr[0].length;j++){
        if(arr[i][j]==1){
            arr[0][j]=1;
            arr[i][0]=1;
        }
    }
}
```

O(n²)

```
for(int i=1;i<arr.length;i++){
    for(int j =1;j<arr[0].length;j++){
        if(arr[0][j]==1 || arr[i][0]==1){
            arr[i][j]=1;
        }
    }
}
```

O(n²)

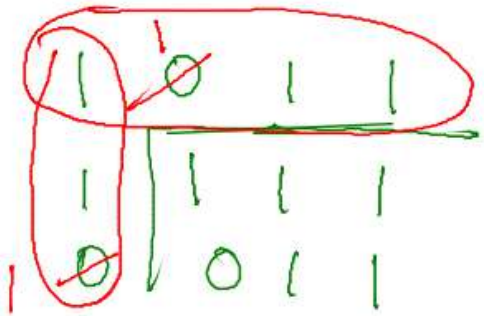



```
  if(firstrowFlag){
      for(int j=0;j<col;j++){
          arr[0][j]=1;
      }
  }

  if(firstcolFlag){
      for(int i=0;i<row;i++){
          arr[i][0]=1;
      }
  }
}
```

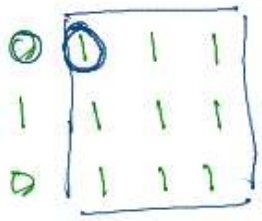
O(n)

O(n)



Ques

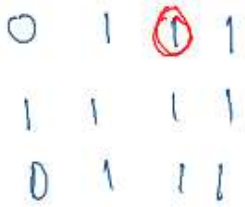
2D =



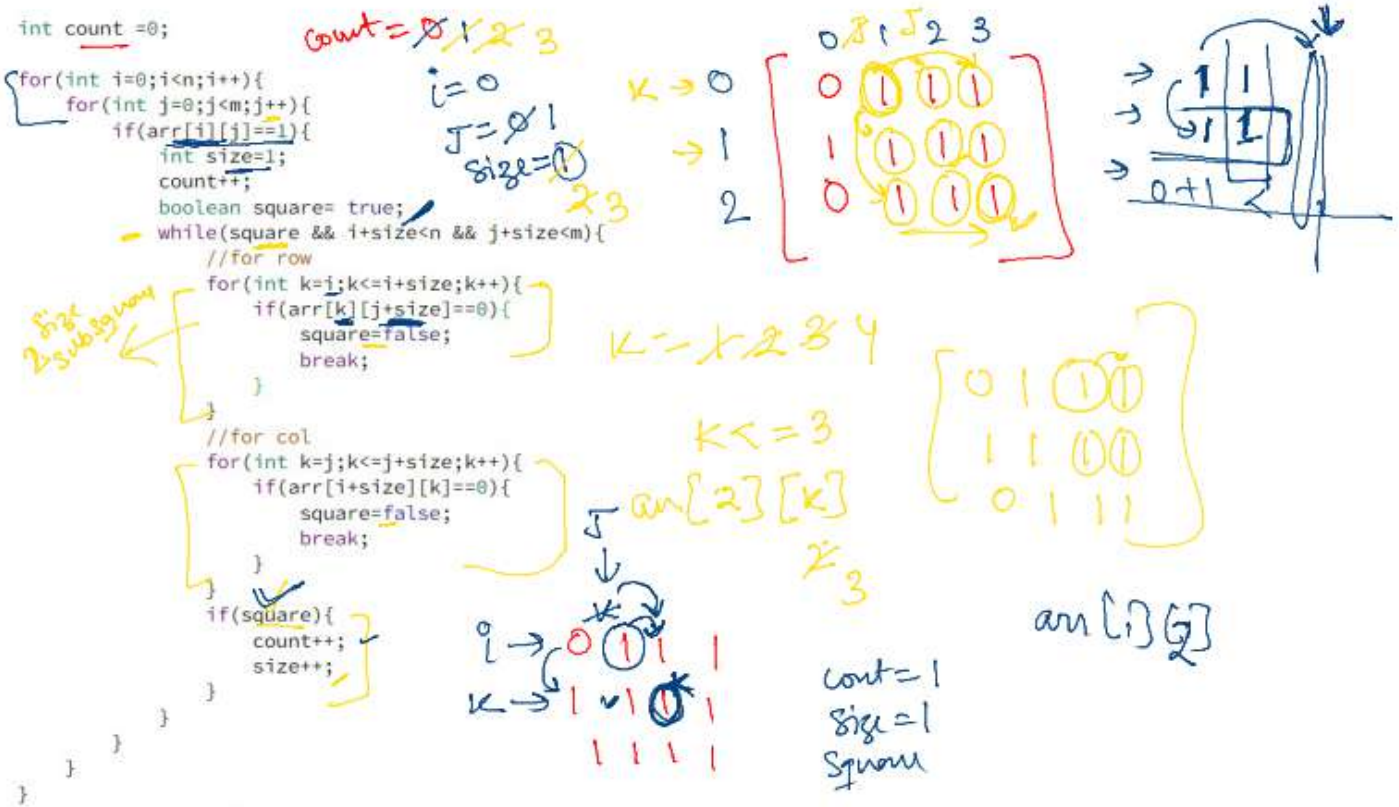
# of square 1's array

size → 1 → 10
→ 2 → 4 + ⇒ 15 subsquare matrix
→ 3 → 1 + with all 1's

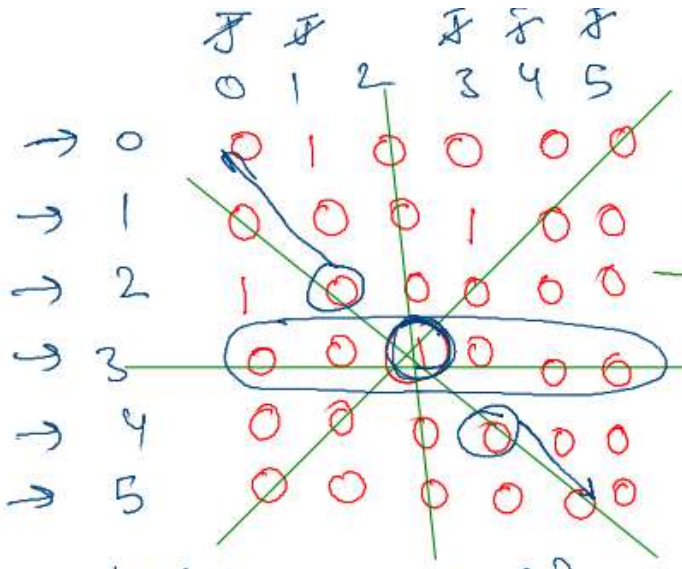
if(arr[i][j]==1)



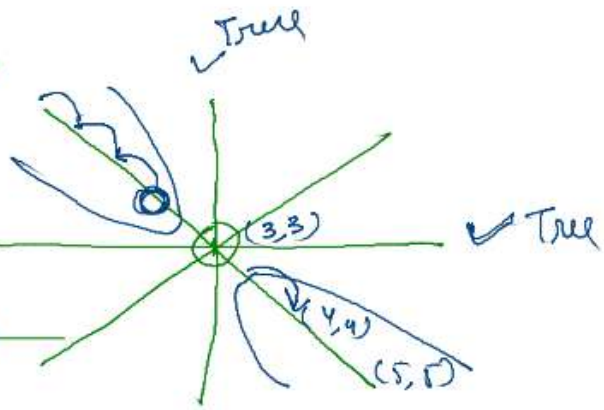
```
int count =0;
for(int i=0;i<n;i++){
    for(int j=0;j<m;j++){
        if(arr[i][j]==1){
            int size=1;
            count++;
            boolean square= true;
            while(square && i+size<n && j+size<m){
                //for row
                for(int k=j;k<=i+size;k++){
                    if(arr[k][j+size]==0){
                        square=false;
                        break;
                    }
                }
                //for col
                for(int k=j;k<=j+size;k++){
                    if(arr[i+size][k]==0){
                        square=false;
                        break;
                    }
                }
                if(square){
                    count++;
                    size++;
                }
            }
        }
    }
}
System.out.println(count);
```



Qus:



issafe(arr, 3, 2)
row col



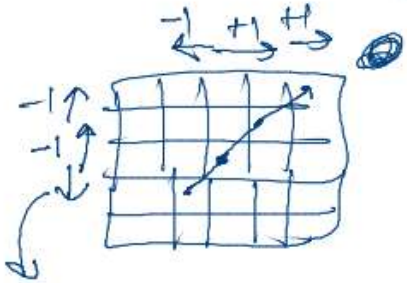
i-1, j-1

// vertical

```
for(int i=0 ; i<n ;i++){
if( i!=row && arr[i][z]==1){
        return false
}
}
```

// horizonal

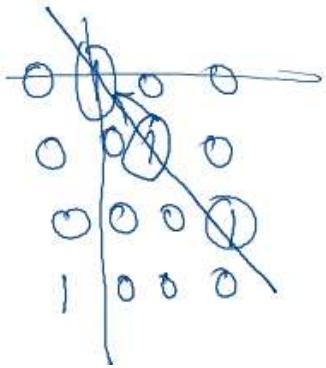
[arr[col][i]==1]

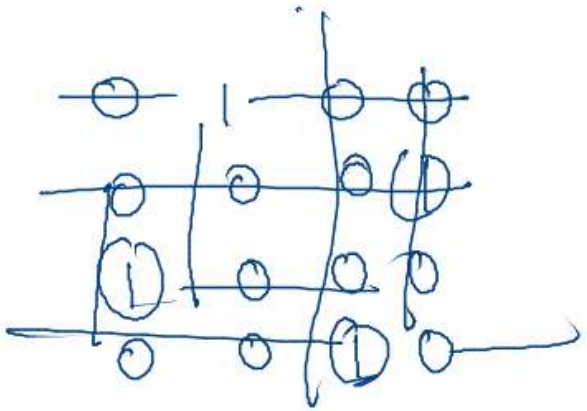


arr[0].len

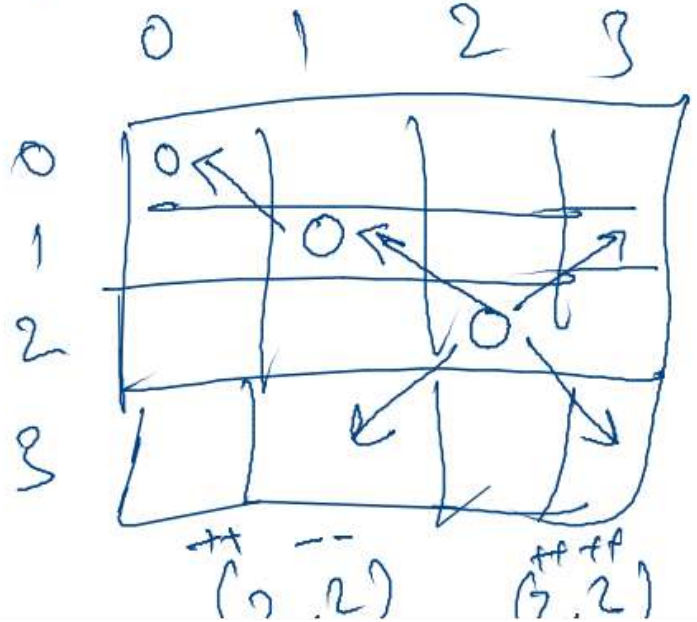
j>=0

i<arr.n





nqueen



danger

-- ++

(2,2)

(1,3)

(2,2)

(1,1)

(0,0)

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[][] arr= new int[n][n];
    for(int i=0;i<n;i++){
        for(int j =0;j<n;j++){
            arr[i][j]= scn.nextInt();
        }
    }

    solution(arr);
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be r
}

public static void solution(int[][] arr){
    boolean flag=false;
    for(int i=0;i<arr.length;i++){
        for(int j=0;j<arr[0].length;j++){
            if(arr[i][j]==1){
                if(isSafe(arr,i,j)==false){
                    flag=true;
                    break;
                }
            }
        }
    }

    if(flag){
        System.out.println("Danger");
    }else{
        System.out.println("N Queens");
    }
}
```

```java
public static boolean isSafe(int[][] arr, int row, int col){
    //vertical check
    for(int i=0;i<arr.length;i++){
        if(i!=row && arr[i][col]==1){
            return false;
        }
    }

    //horizontal check
    for(int j=0;j<arr[0].length;j++){
        if(j!=col && arr[row][j]==1){
            return false;
        }
    }

    //right diagonal check
    int i= row-1;
    int j = col-1;
    while(i>=0 && j>=0){
        if(arr[i][j]==1){
            return false;
        }
        i--;
        j--;
    }

    i= row+1;
    j = col+1;
    while(i<arr.length && j<arr[0].length){
        if(arr[i][j]==1){
            return false;
        }
        i++;
        j++;
    }
```

```
    //left diagonal
    i= row-1;
    j = col+1;
    while(i>=0 && j<arr[0].length){
        if(arr[i][j]==1){
            return false;
        }
        i--;
        j++;
    }
    i= row+1;
    j = col-1;
    while(j>=0 && i<arr.length){
        if(arr[i][j]==1){
            return false;
        }
        i++;
        j--;
    }
    return true;
}
```