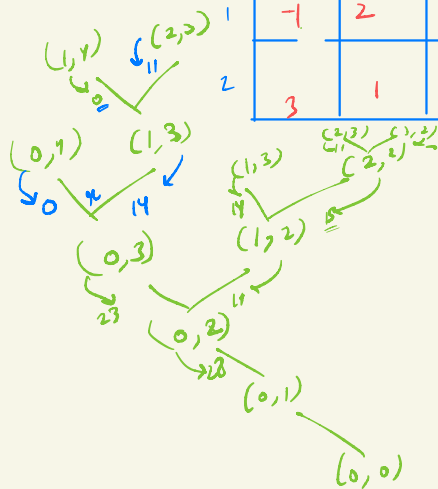



(i, j)



	0	1	2	3
0	1	0	5	9
1	-1	2	4	3
2	3	1	4	11

max path sum in matrix from $(0,0)$ to $(n-1, m-1)$

```
public static int maxSum(int i, int j, int[][] matrix, int n, int m) {
    if(i >= n || j >= m || i < 0 || j < 0) {
        return 0;
    }

    if(i == n-1 && j == m-1) {
        return matrix[i][j];
    }

    // right direction
    int a = maxSum(i, j+1, matrix, n, m);

    // downwards direction
    int b = maxSum(i+1, j, matrix, n, m);

    int ans = Math.max(a, b) + matrix[i][j];

    return ans;
}
```

~~flood fill~~

0	1	0	1	1	1
0	1	0	1	1	0
0	1	0	0	0	0
0	0	0	1	1	0
0	1	0	1	1	0
0	0	0	1	1	0

```
public static void floodFill(int i, int j, int[][] mat, int n, int m) {
    if(i < 0 || j < 0 || i >= n || j >= m || mat[i][j] == 1 || visited[i][j]) {
        return;
    }

    if(i == n-1 && j == m-1) {
        System.out.println(psf);
        return;
    }

    visited[i][j] = true;
    floodFill(i-1, j, mat, n, m, psf+"t", visited);
    floodFill(i, j-1, mat, n, m, psf+"l", visited);
    floodFill(i+1, j, mat, n, m, psf+"d", visited);
    floodFill(i, j+1, mat, n, m, psf+"r", visited);
    visited[i][j] = false;
}
```

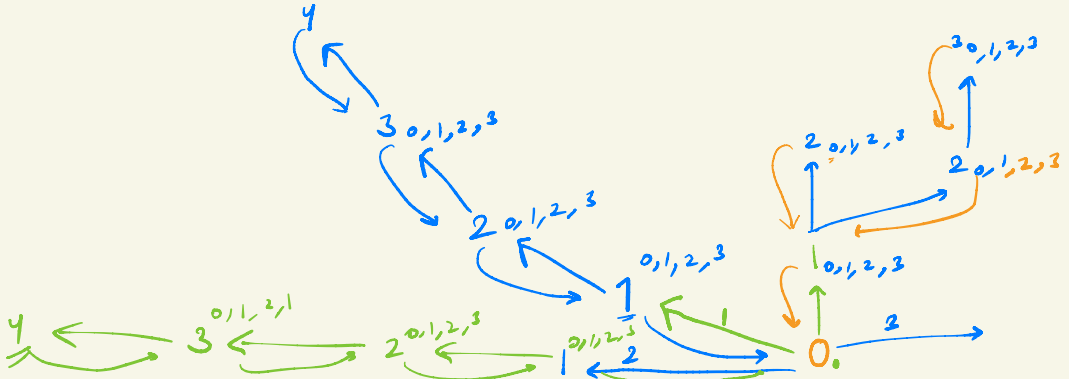
break test
10:15

every row
will have
exactly 1 queen
(0,1) (1,3) (2,0) (3,2)
(0,2) (1,0) (2,3) (3,1)

	0	1	2	3
0				Q
1				
2				
3				

```
public static void NQueens(int row, int n, boolean[][] board, String asf){
    if(row==n){
        System.out.println(asf);
        return;
    }

    for(int col=0; col<n; col++){
        if(isQueenSafe(row,col,board)==true){
            board[row][col] = true;
            NQueens(row+1, n, board, asf+" "+col+" ");
            board[row][col] = false;
        }
    }
}
```



int C() dir = { -1, -1, -1, 0, -1, 1, 1, 0, 1, 1, 1, 0, 1, -1, 0, -1, -1, -2, -2, 0, -2, 2, 2, 2, 2, 0, 3, -2, -2, -3, -3, 0, 3, -3, 3, 0, 3, 3, 3, 3, 0, 3, -3, 3, -3 }

	0	1	2	3	4	5
0						
1						
2						
3						
4						

row=2
col=3

dir = 6, 2, 2

rad=2
dir = (-1, 0)
i = 2 + 2 * (-1) = 2 - 2 = 0
j = 3 + 2 * (0) = 3 + 0 = 3

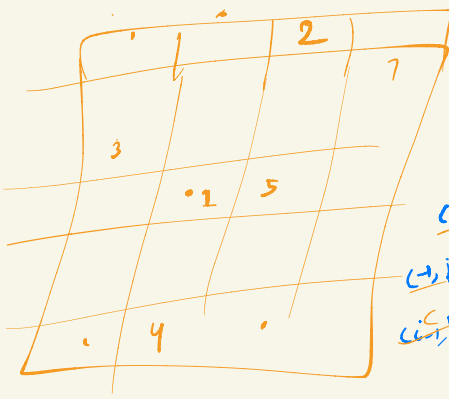
```
public static boolean isQueenSafe(int row, int col, boolean[][] board){
    int n = board.length;

    int[][] dirs = {{-1,-1},{-1,0},{-1,1},{0,1},{1,1},{1,0},{1,-1},{0,-1}};

    for(int rad=1; rad<n; rad++){
        for(int dir : dirs){
            int i = row + rad * dir[0];
            int j = col + rad * dir[1];

            if(i>=0 && j>=0 && i<n && j<n && board[i][j]==true) return false;
        }
    }

    return true;
}
```

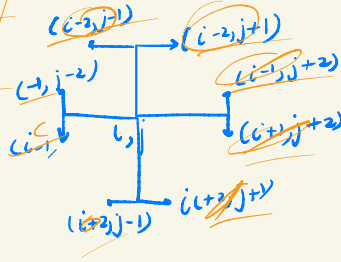


8x8

if (move == 2)
return 20; else

chess(w)(c) = move;

move + 1



chess(w)(c) = 0;