

```

...

    Time complexity:  $O(N*M)$ 
    Space complexity:  $O(N*M)$ 
    where N and M are the matrix parameters

...

from sys import stdin
dir = [[1, 0], [-1, 0], [0, 1], [0, -1]]

def valid(x, y, n) :

    return (x >= 0 and y >= 0 and x < n and y < n)

def __dfs(edge, vis, x, y, n) :

    vis[x][y] = True
    count = 1

    for i in range(4) :

        nex = x + dir[i][0]
        ney = y + dir[i][1]

        if(valid(nex, ney, n) and edge[nex][ney] == '1' and vis[nex][ney] == False) :

            count += __dfs(edge, vis, nex, ney, n)

    return count

def dfs(edge, n) :

    vis = [[False for i in range(n)] for j in range(n)]
    ans = 0

    for i in range(n) :

        for j in range(n) :

            if(vis[i][j] == False and edge[i][j] == '1') :

                ans = max(ans, __dfs(edge, vis, i, j, n))

    return ans

# Main

n = int(stdin.readline().strip())

```

```
edge = [stdin.readline().strip().split() for i in range(n)]  
print(dfs(edge, n))
```