Ejercicio 8 *(CazadorDeFalsos)* ★

Se tiene una matriz booleana A de $n \times n$ y una operación *conjunciónSubmatriz* que toma $O(1)$ tiempo y que dados 4 enteros i_0, i_1, j_0, j_1 devuelve la conjunción de todos los elementos en la submatriz que toma las filas i_0 hasta i_1 y las columnas j_0 hasta j_1. Formalmente:

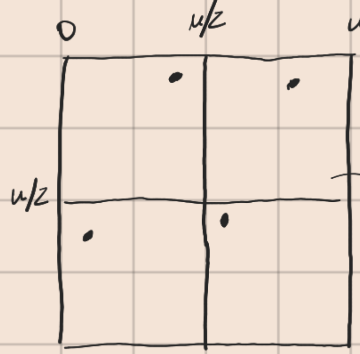
$$\text{conjunciónSubmatriz}(i_0, i_1, j_0, j_1) = \bigwedge_{i_0 \leq i \leq i_1, j_0 \leq j \leq j_1} A[i, j]$$

1. Dar un algoritmo de complejidad temporal estrictamente menor que $O(n^2)$ que calcule la posición de algún false, asumiendo que hay al menos uno. Calcular y justificar la complejidad del algoritmo.

2. Modificar el algoritmo anterior para que cuente cuántos false hay en la matriz. Asumiendo que hay a lo sumo 5 elementos false en toda la matriz, calcular y justificar la complejidad del algoritmo. Esto se puede lograr con complejidad menor a $O(n^2)$.

1. Sabemos que hay al menos un false en mi matriz → puedo dividir a mi matriz en 4 submatrices y el false va a estar ahí, puedo descartar las otras 3

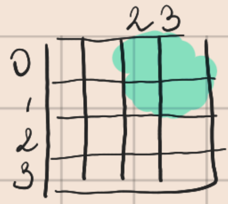
solo necesito la pos de 1 ← puede haber + de 1 ←



→ puedo ir dividiendo mi matriz en 4 y voy preguntando donde hay 1 y busco en el cuadrante que 1ro me haya dado false

↳ Caso base: matriz 1×1 → Conquer

↳ Combine: no tiene :)




```
PosFalse(M, i0, i1, j0, j1)
    if i0 = i1 ^ j0 = j1
        if !CS(i0, i1, j0, j1)
            return (i0, j0)
    else
        midW = ⌊(i1 - i0)/2⌋ , midh = ⌊(j1 - j0)/2⌋        necesario sumar → pensar
                                                            i0 = 2   i1 = 4
        if !CS(M, i0, midW+i0, j0, midh+j0)                 ||       ||
                                                            j0       j1
            return PosFalse(M, i0, midW, j0, midh)      →   1er cuadrante
        if !CS(M, midW+1+i0, i1, j0, midh+j0)
            return PosFalse(M, midW+1+i0, i1, j0, midh+j0)  →   2do cuadrante
        if !CS(M, i0, midW+i0, midh+1+j0, j1)
            return PosFalse(M, i0, midW+i0, midh+1+j0, j1)  →   3er cuadrante
        if !CS(M, midW+1+i0, i1, midh+1+j0, j1)
            return PosFalse(M, midW+1+i0, i1, midh+1+j0, j1)  →   4to cuadrante
```

$T(n) = T(n/4) + \Theta(1)$ → Como n = alto → cuando tomo un cuadrante es $\frac{n}{2}$

$a = 1, b = 4 \rightarrow \log_b a = 0 \rightarrow \Theta(1) \in \Theta(n^0)$? Sí → $T(n) = \Theta(\log n)$

$T(n) = T(n/2) + \Theta(1) = \Theta(n)$

CountFalse $(M, i_0, i_1, j_0, j_1$

if $i_0 = i_1 \wedge j_0 = j_1$

if $!CS(i_0, i_1 j_0, j_1)$

return 1

else

return 0

else

$midW = \lfloor (i_1 - i_0)/2 \rfloor$, $midh = \lfloor (j_1 - j_0)/2 \rfloor$

c_1 += CountFalse $(M, i_0, midW, j_0, midh)$ → 1^{er} cuadrante

c_2 += CountFalse $(M, midW+1+i_0, i_1, j_0, midh+j_0)$ → 2^{do} cuadrante

c_3 += CountFalse $(M, i_0, midW+i_0, midh+1+j_0, j_1)$ → 3^{er} cuadrante

c_4 += CountFalse $(M, midW+1+i_0, i_1, midh+1+j_0, j_1)$ → 4^{to} cuadrante

return $c_1 + c_2 + c_3 + c_4$

$T(n) = 4T(n/4) + \Theta(1)$ ✗ $T(n) = 4T(n/2) + \Theta(1) = O(n^2)$

$a = 4, b = 4$ → $\log_b a = 1$

¿$\Theta(1) \in O(n^{1-\varepsilon})$ para $\varepsilon > 0$? Sí, sea $\varepsilon = 1/2$ vale

→ x caso 1 TM $T(n) = \Theta(n)$