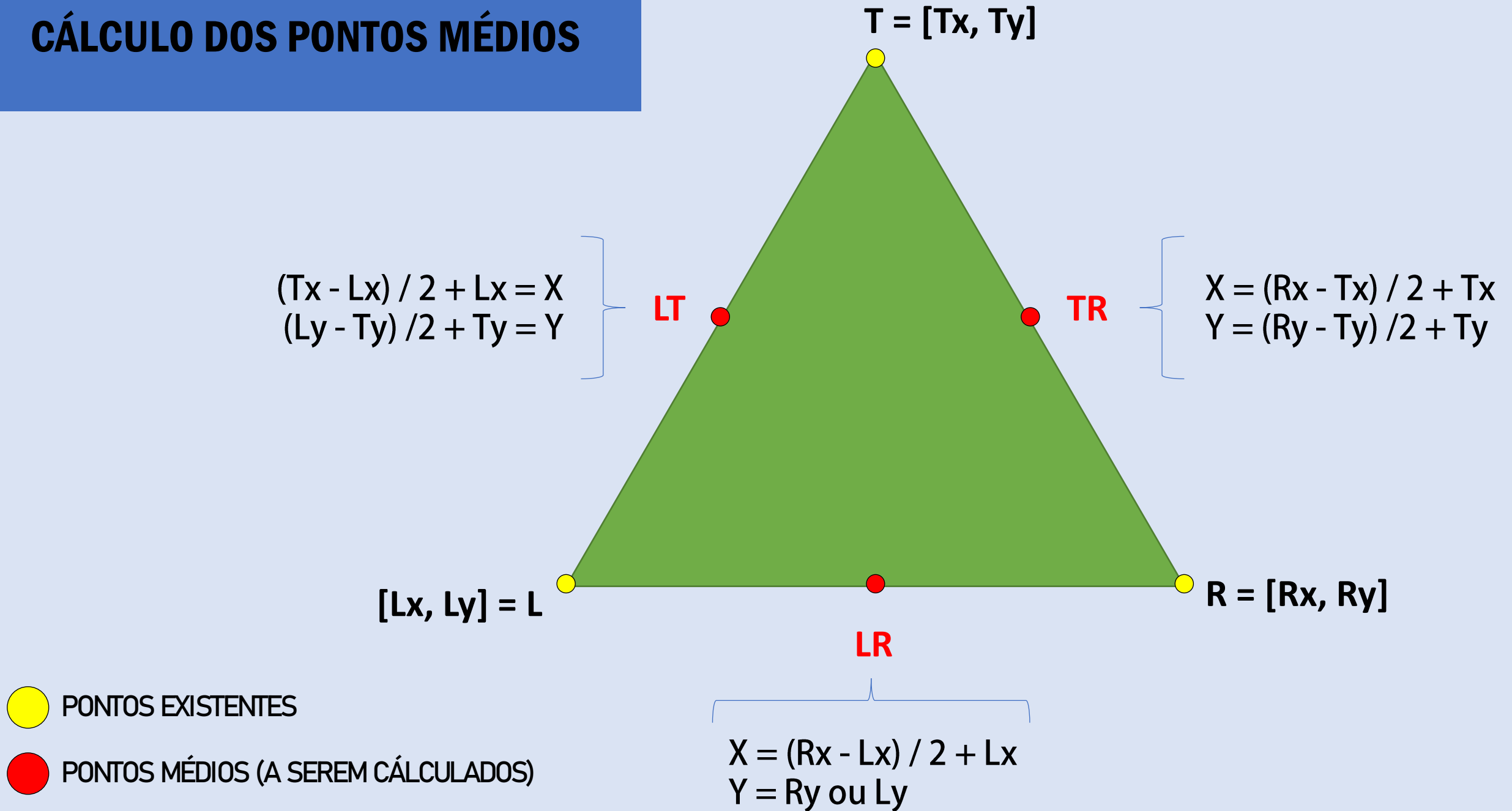


A decorative border at the top of the slide featuring a repeating pattern of black and orange triangles, forming a Sierpinski triangle fractal.

0 triângulo de Sierpiński

A decorative border at the bottom of the slide, identical to the top one, featuring a repeating pattern of black and orange triangles forming a Sierpinski triangle fractal.

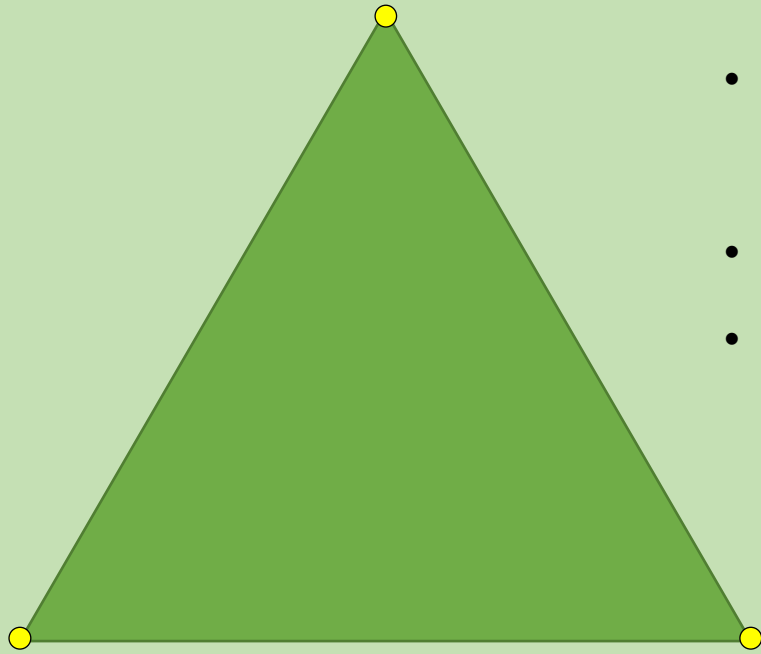
CÁLCULO DOS PONTOS MÉDIOS





TRIANGULO

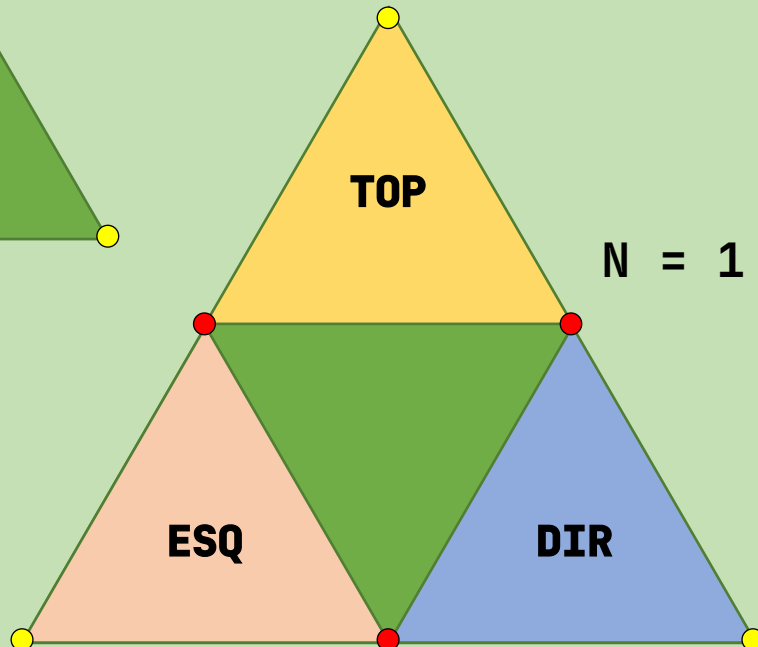
PASSOS DO ALGORÍTMO:

- RECEBE 3 PONTOS REFERENTE A UM TRIANGULO
- É FEITO O CÁLCULO DOS PONTOS MÉDIOS
- A PARTIR DOS PONTOS MÉDIOS CONSEGUIMOS DIVIDIR O TRIANGULO EM 3 E ITERAR RECURSIVAMENTE.
- A RECURSÃO SÓ PARA QUANDO n FOR 0.
- QUANDO N FOR 0, EXECUTA `POINTS.GETPOINTS()`

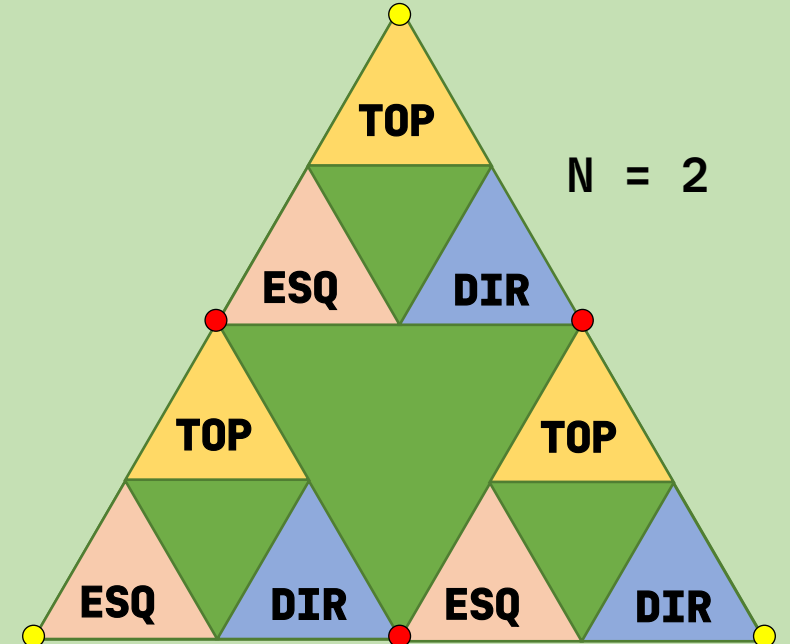


$N = 0$

-  PONTOS EXISTENTES
-  PONTOS MÉDIOS (A SEREM CÁLCULADOS)



$N = 1$



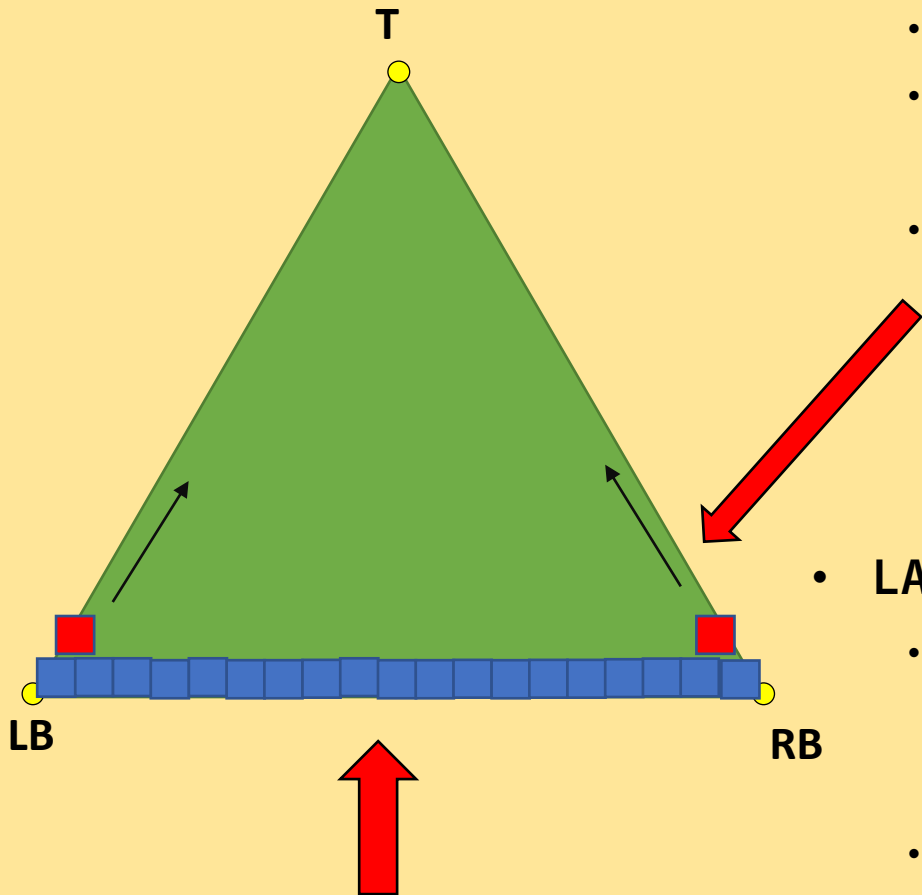
$N = 2$

POINTS

PASSOS DO ALGORÍTMO:

- **GETPOINTS:**

- RECEBE OS 3 PONTOS $[x,y]$ REFERENTE AO TRIANGULO.
- UTILIZA DA FUNÇÃO **LATERALPOINTS()** INSERINDO O x DE LB E O x de RB, OBTEM TODOS OS VALORES DE x ENTRE ELES.
- É MODIFICADO OS PONTOS $[x,y]$ DE LB E RB NA RECURSÃO ATÉ QUE y DE LB (OU RB, TANTO FAZ) SEJA IGUAL A y DE T, OU SEJA, ESTEJAM NA MESMA ALTURA.



VALORES DE x OBTIDOS POR
LATERALPOINTS()

- **LATERALPOINTS:**

- RECEBE O OFFSET (INICIO, x de LB) E LIMIT (x DE RB), E TAMBÉM RECEBE y PARA ENTREGAR COM OS PONTOS JÁ FORMATADOS: $[x, y]$.
- RETORNAR TODOS OS PONTOS $[x,y]$ DAQUELA LINHA y JÁ CONCATENADOS.
- A RECURSÃO PARA QUANDO $OFFSET == LIMIT$.

MATRIX

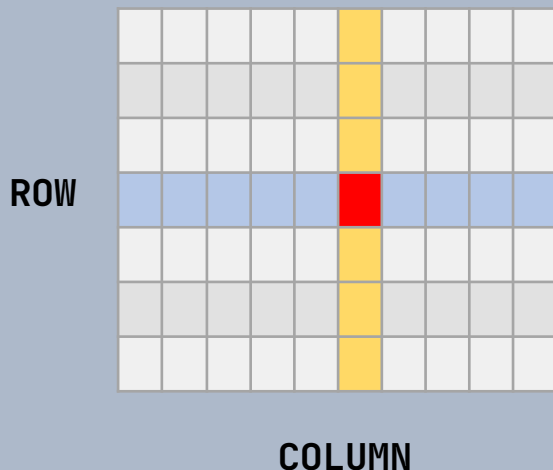
PASSOS DO ALGORÍTMO:

- **MODIFY:**

- MÉTODO RESPONSÁVEL POR EDITAR UMA POSIÇÃO **[x,y]** NA MATRIZ POR UM VALOR **value**.
- ELA RECEBE A MATRIZ, LINHA, COLUNA E VALOR A SER INSERIDO NO LOCAL. ENVIA ESSES DADOS PARA A FUNÇÃO **ROW()**.

- **ROW:**

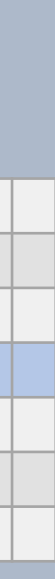
- ESSE MÉTODO É RESPONSÁVEL POR PERCORRER AS LINHAS DO ARRAY. ELA RECEBE TODOS OS DADOS RECEBIDOS EM **MODIFY()**, E MAIS UM ATRIBUTO **N_ROW** INICIALIZADO COM **1** PARA INDICAR QUAL LINHA SE ENCONTRA.
- OBTEM A PRIMEIRA LINHA COM **[head | body] = matrix**.
- VERIFICA-SE SE ESTÁ NA LINHA CORRETA COM O IF, SE A LINHA ATUAL É IGUAL A LINHA DE INTERESSE, OU SEJA: **n_row == row_edit**.
 - **SE NÃO:** A HEAD VAI SER A MESMA, OU SEJA, MESMA LINHA.
 - **SE SIM:** A HEAD DE RETORNO SERÁ UMA LINHA NOVA COM O VALOR MODIFICADO, E É FEITO PELA FUNÇÃO **COLUMN()**.



MATRIX

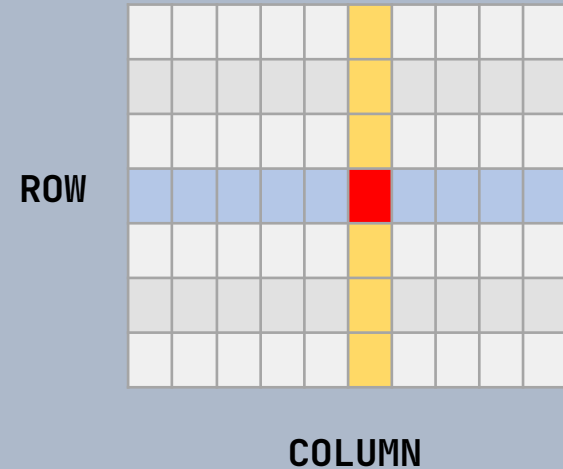
PASSOS DO ALGORITMO:

- **COLUMN:**

- 
- MÉTODO RESPONSÁVEL POR PERCORRER AS COLUNAS. ELA RECEBE A **LINHA**, UM **N_COL** QUE É INICIALIZADO COM 1 E IDENTIFICA EM QUAL COLUNA INICIAR, **COL_EDIT** QUE É A COLUNA QUE DEVE SER EDITADA E O **VALUE**.
 - SEMELHANTE A **ROW()**, ELA PERCORRE AS COLUNAS VERIFICANDO COM UM **IF** SE ESTÁ NA COLUNA DE INTERESSE, OU SEJA, **n_col == col_edit**.
 - SE NÃO, O **HEAD(VALOR)** CONTINUA O MESMO.
 - SE SIM, O **VALUE** É COLOCADO NO LUGAR DO VALOR ANTERIOR.
 - RECURSIVAMENTE RETORNA A LINHA, PORÉM, COM O VALOR DA COLUNA MODIFICADO.

- **PRINT:**

- IMPRIME A MATRIZ DE TRIANGULOS



CONVERT

PASSOS DO ALGORÍTMO:

- **GETFINALMATRIX:**

- ESSA FUNÇÃO RECEBE A MATRIZ EM BRANCO (TRACEJADA) E TODOS OS PONTOS **[x,y]** QUE DEVEM SER PINTADOS (COLOCADO "0"). ELE É RESPONSÁVEL POR PEGAR A MATRIZ EM BRANCO E RETORNAR UMA MATRIZ COM OS TRIANGULOS.
- ELA PEGA CADA UM DOS PONTOS **[x,y]**, USANDO **MATRIX.MODIFY()** MODIFICA A POSIÇÃO DA MATRIZ EM BRANCO NOS PONTOS **x** e **y** INDICADOS.
- DEPOIS ELA ENTRA NA RECURSÃO COM ESSA NOVA MATRIZ E COM O RESTANTE DOS PONTOS QUE FALTAM MODIFICAR.
- A RECURSÃO É FEITA ATÉ A LISTA DE POSIÇÕES **[x,y]** ACABAR, OU SEJA, **length(positions) == 0**.

ROW

COLUMN



FIM

ELLISON WILLIAM

GEORGE LUCAS (VIN DIESEL DE CAMACAN)

JÂNIO JUNIOR

