TALLER 1 – JJJS

JUAN JOSE GIRALDO

JUAN MANUEL GOMEZ

JUAN CAMILO MANJARRÉS

SAMUEL VALENCIA

PROFESOR:

JUAN CAMILO DAVID DiAZ

ANÁLISIS Y DISEÑO DE ALGORITMOS

INGENIERÍA DE SISTEMAS

UNIVERSIDAD DE MEDELLÍN

MEDELLIN

2023-1

[**Recorrido zig-zag**](#_fhnq5y7zx51e) **2**

[**Recorrido diagonal**](#_h1680vh0psoh) **2**

[Recursivo](#_2lct5g7yf10b) 2

[No recursivo](#_bi87udnkwqx5) 3

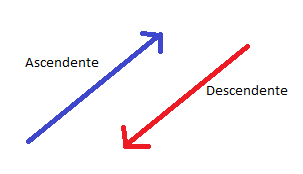
[**Referencias**](#_7vst320o00w) **4**

# 

# Recorrido diagonal secundaria

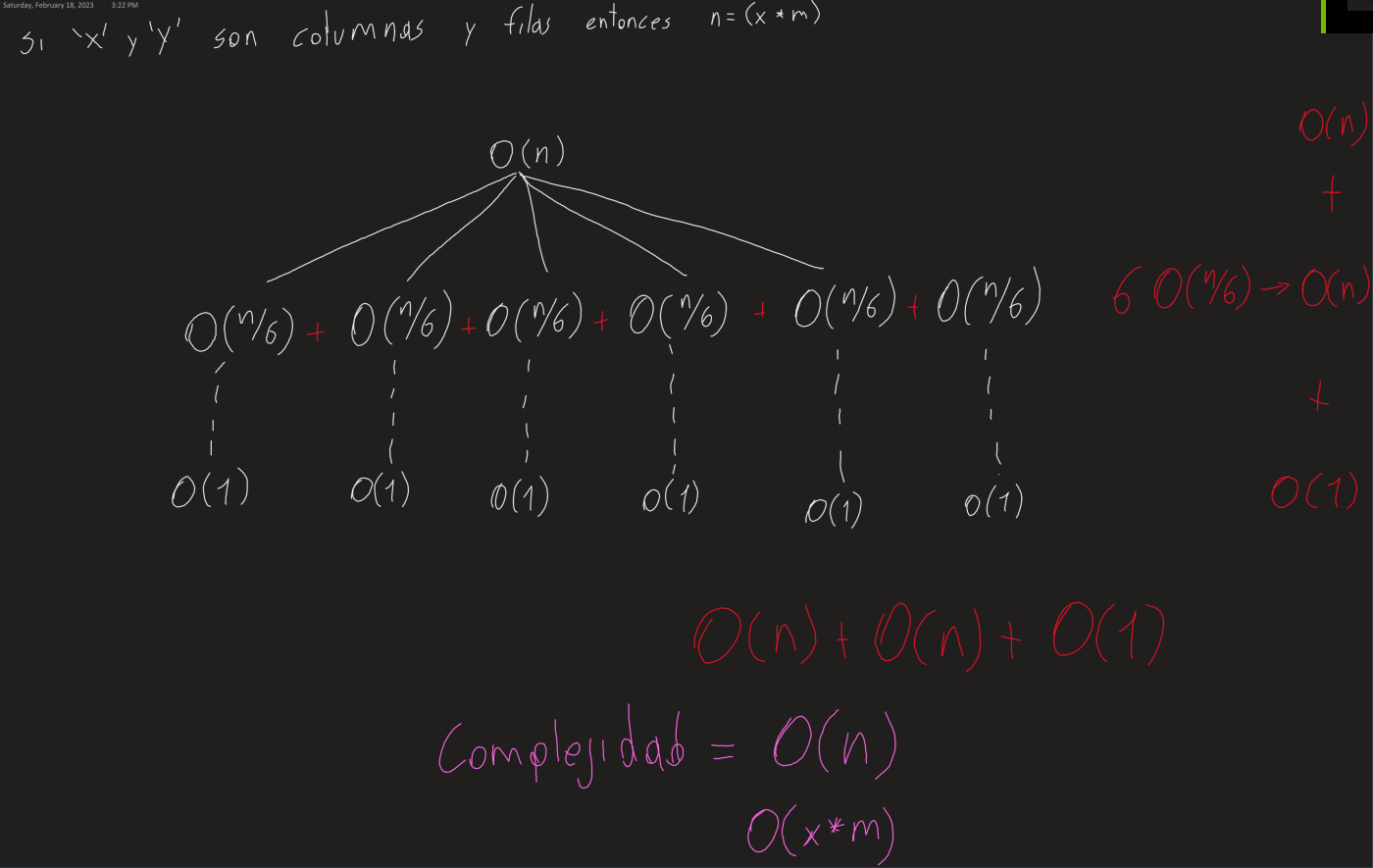
## Recursivo

La función recibirá la columna, fila y dirección actual y dependiendo de la dirección (ascendente o descendente) las variables columnas o fila se modificarán.



Después se imprime la celda respectiva a la columna y fila actual que se encuentre.

La complejidad se calculó con el método del árbol recursivo.

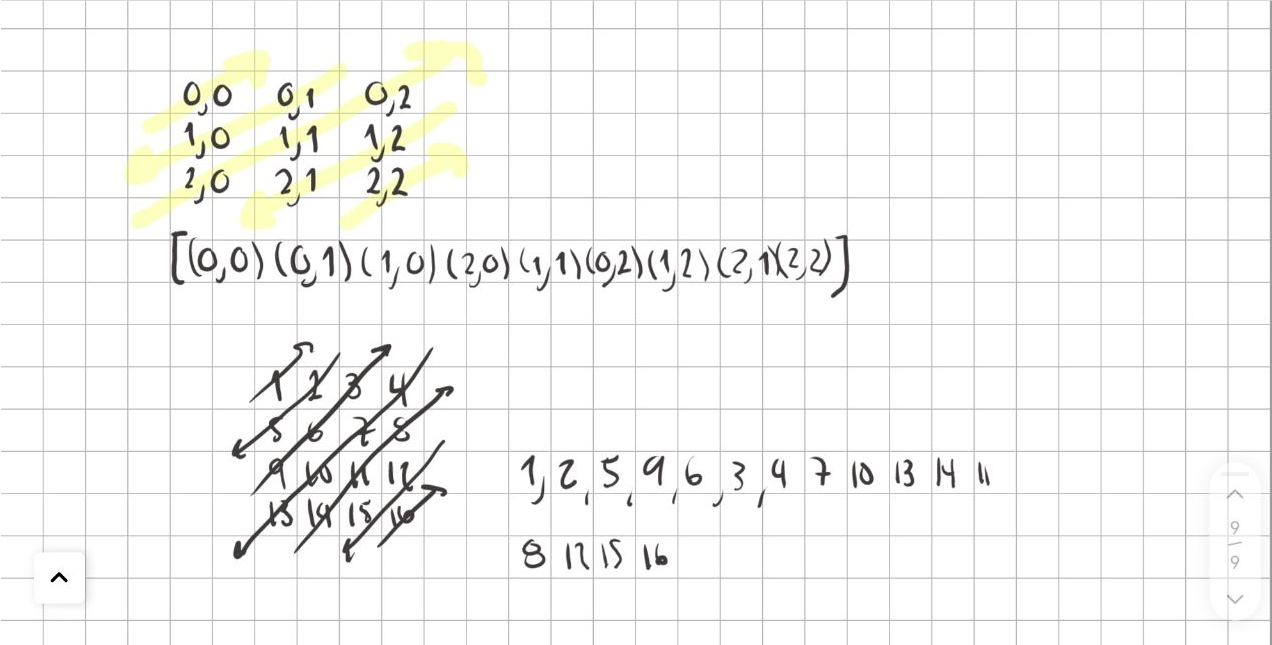


Tenemos una función que recibe n entradas y en esa misma función hay 6 llamadas recursivas, entonces el problema se divide en 6 subproblemas. Entonces la ecuación iniciaría con T(n) = O(n/6) + …

Cada una de esas n/6 van a tener otras n/6. Entonces cuando cada generación del árbol se suma, dará O(n). Después se suman los resultados de todas las generaciones O(n)+O(n)+...

## No recursivo

La variable dirección será utilizada para determinar si será una diagonal ascendente o descendente.



Se inicializan las variables columna y fila para ubicarse en la matriz.

En el ‘for’ se manipulan las filas y las columnas dependiendo si es ascendente y descendente, y el ‘print’ al inicio del ‘for’; expondrá de acuerdo a la fila y columna actual.

La complejidad fué calculada teniendo en cuenta que las operaciones de una sola ejecución o fuera del ciclo ‘ for‘ ocuparan una complejidad de O(1). Las líneas de código dentro del ciclo tendrán la complejidad O(n) incluyendo las que contienen los ‘if’ dentro del ‘for’.

# 

# Referencias

* <https://www.geeksforgeeks.org/zigzag-or-diagonal-traversal-of-matrix/>
* https://www.youtube.com/watch?v=dqFS-CXCEVQ&ab\_channel=ChioCode