Documento Técnico PRE

PROBLEMA I

ELABORADO POR: JUAN CAMILO ROJAS CATRO

PRESENTADO A: SESBASTIAN CAMILO MARTINEZ REYES

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

AYED

PROGRAMA DE INGENIERÍA DE SISTEMAS

BOGOTÁ D. C.

31 DE AGOSTO DE 2020

**1. Especificación:**

* Entrada:

La entrada es básicamente la cantidad de tableros que se van a ingresar seguida de todos los diferentes tableros.

La salida es un si o no para cada tablero, en caso de que pueda o no ser el tablero final de un juego de Tic tac toe.

**2. Estrategia:**

* Descripción y justificación**:**

Para facilitar el conteo de X, O y encontrar el ganador lo más sencillo es convertir en números los espacios vacíos las X y las O, una vez tenemos esto procedemos a verificar si cumple las reglas.

* Estructura de datos:

Para el problema solo se van a utilizar enteros, y listas bidimensionales.

* Algoritmo:

Para la solución tenemos 3 programas, el que lee el tablero y lo pasa a una matriz, el que transforma la matriz a números y el que verifica el tablero y encuentra el ganador, con las siguientes condiciones:

- Solo puede haber un ganador.

- Máximo un solo jugador puede ganar dos veces por tablero.

- No puede haber más O que X.

- Si las O ganaron hay igual cantidad de O que de X.

- Solo puede haber una X más que O’s.

- Si ganan las X hay más X’s.

- La dimensión del tablero tiene que ser 3x3.

**3. Casos de prueba:**

* Entrada:

El archivo de texto adjunto como input\_I.txt tiene exactamente 500 casos de prueba totalmente diferentes.

* Salida:

El archivo de texto adjunto como output\_I.txt tiene cada la salida de cada caso ingresado.

* Justificación:

Esta cantidad de casos es para verificar que el algoritmo no tiene ningún problema, o caso excluido que pueda generar una salida errónea.

**4. Análisis:**

* Temporal:

Debido a la cantidad de ciclos anidados la complejidad es O(n^2)

**5. Documentación:**

* El nombre del código respectivo es “Problem\_I.py”