Análisis de Eficiencia del Código del Sistema de Árbol Genealógico

Este documento analiza la eficiencia del sistema de árbol genealógico desarrollado en Python, incluyendo sus principales operaciones y complejidad computacional. El sistema permite gestionar relaciones familiares, como padres, hermanos, parejas y mascotas, y utiliza estructuras de datos adecuadas para mantener y visualizar el árbol genealógico.

Análisis de Eficiencia

1. Estructura de Datos Utilizada

La estructura principal utilizada es un diccionario (hash map) para almacenar cada persona por su ID único.

Esto permite un acceso O(1) en promedio a los datos de cada persona, facilitando operaciones de búsqueda y modificación en el árbol.

2. Complejidad de las Operaciones Principales

- Agregar Persona: O(1) para agregar una persona al diccionario. La asignación de padres/madres implica también O(1), ya que se hace a través de referencias directas en los atributos de la persona.

- Agregar Hermano/Hermana: O(1) para encontrar al hermano existente en el diccionario. Si se asignan nuevos padres, también es una operación O(1), dado que no se requiere una búsqueda adicional, sino solo referencias directas.

- Asignar Pareja: O(1) en promedio para encontrar ambas personas en el diccionario y establecer la relación.

- Agregar Mascota: O(1) para agregar una mascota a la lista de mascotas de una persona. Las mascotas están asociadas directamente a su dueño, por lo que no necesitan una estructura de datos adicional.

3. Complejidad de Visualización

La visualización del árbol genealógico se realiza con la biblioteca `networkx` y `matplotlib`, que tienen una complejidad de representación aproximadamente O(n), donde n es el número de nodos en el árbol (personas y mascotas). La complejidad aumenta en función de la cantidad de conexiones y el tamaño del árbol.

4. Consumo de Memoria

- Diccionario de Personas: Cada persona tiene referencias a sus padres, hijos y pareja, lo que consume memoria lineal O(n).

- Mascotas: Cada persona tiene una lista de mascotas, que incrementa el consumo de memoria, pero en una escala manejable, ya que cada persona almacena solo sus mascotas.

5. Consideraciones de Optimización

Dado que el árbol genealógico es pequeño en términos de datos (número de personas y mascotas), las operaciones mantienen una buena eficiencia. Para árboles de gran tamaño, una optimización futura podría ser el uso de una base de datos para almacenar relaciones complejas y realizar consultas más avanzadas.

6. Escalabilidad

La estructura actual es escalable para familias pequeñas a medianas. Si se expande a un árbol con cientos de relaciones, es recomendable optimizar la visualización y considerar el uso de un sistema de almacenamiento más avanzado.