

Reflexión act 2.3

Este código aborda el problema de Josephus, que es un problema clásico de teoría de números y estructuras de datos. El objetivo es determinar la posición segura desde la cual comenzar a eliminar personas en un círculo para que, siguiendo un patrón de eliminación específico, quede una sola persona en pie al final.

El enfoque del código es utilizar una estructura de datos de lista enlazada para representar el círculo de personas. Cada persona en el círculo se identifica con un número y se almacena en la lista. El programa simula el proceso de eliminación al contar y eliminar personas en cada ronda hasta que solo queda una.

Sin embargo, una de las principales características de este código es su complejidad. El bucle while anidado dentro de la función josephus resulta en una complejidad de $O(n^2)$, donde n es el número de personas en el círculo. Esto se debe a que, en cada iteración del bucle while, se realiza un bucle for que itera $k-1$ veces para encontrar la siguiente persona a eliminar. Como resultado, el número total de operaciones se incrementa cuadráticamente a medida que n aumenta.

Para mejorar la eficiencia del código, sería beneficioso considerar una estructura de datos o un enfoque algorítmico diferente. Por ejemplo, utilizar un arreglo en lugar de una lista enlazada podría reducir la complejidad a $O(n)$, lo que sería más eficiente para casos con un gran número de personas. Además, se pueden eliminar algunas operaciones redundantes, como el ajuste del iterador current, lo que simplificaría el código.

En resumen, este código resuelve el problema de Josephus de manera funcional, pero su eficiencia podría mejorarse mediante la optimización de la estructura de datos y la reducción de operaciones innecesarias. El problema de Josephus es un desafío interesante en la teoría de números y la estructura de datos, y hay múltiples enfoques para resolverlo de manera más eficiente.