

Explicación del Código: Secuencia Favorita

En esta sección explicaremos cómo implementar dos métodos en Java para reconstruir la secuencia original del problema "Secuencia favorita". El proyecto incluye dos enfoques: un método cándido (o ingenuo) y un método óptimo, ambos recibiendo como parámetros:

Un entero n , que indica la cantidad de elementos.

Un arreglo de enteros que representa la secuencia escrita por Policarpa.

1. Método Cándido (Enfoque Básico)

El método cándido implementa la lógica de forma directa, imitando paso a paso cómo deberíamos reconstruir manualmente la secuencia.

1.1 Variables iniciales

- Un arreglo nuevo donde construiremos la secuencia original.
- Una variable pos , inicializada en 0, que indica la posición que estamos rellenando en el nuevo arreglo.

1.2 Lógica del método

1. Se utiliza un ciclo `while` que avanza desde 0 hasta n , indicando las posiciones que deben llenarse en la nueva secuencia.
2. Dentro del `while`, un ciclo `for` recorre la secuencia que Policarpa escribió, de principio a fin.
3. Cuando el índice del `while` coincide con el índice del `for`, analizamos si esa posición es par o impar:
 - Si es par: tomamos elementos desde el inicio de la secuencia de Policarpa.
 - Si es impar: tomamos elementos desde el final.
4. Cada número seleccionado se coloca en la posición pos del nuevo arreglo.
5. La variable pos incrementa de uno en uno hasta completar la secuencia.

1.3 Resultado

Al ejecutar este método con la secuencia del ejemplo, se obtiene: 9 1 7 2

La cual es la reconstrucción correcta.

Aunque este enfoque funciona, realiza más iteraciones de las necesarias, resultando en una complejidad aproximada de $O(n^2)$.

2. Método Óptimo (Two Pointers)

Este método aplica la técnica de dos punteros para optimizar el proceso y reducir drásticamente el número de iteraciones.

2.1 Variables iniciales

Además del nuevo arreglo y la variable pos , se crean dos punteros:

- L , ubicado al inicio del arreglo (0).
- R , ubicado al final ($n - 1$).

Estos punteros representan los extremos desde donde Policarpa fue tomando elementos.

2.2 Lógica del método

El funcionamiento es el siguiente:

1. Se itera mediante un `while` mientras $L \leq R$.
2. En cada iteración:
 - Se toma el número en la posición L y se coloca en el nuevo arreglo.
 - Luego verificamos si todavía queda espacio para insertar un segundo número:
 - Si $pos + 1 < n$, colocamos el elemento en la posición R .
3. Después de ubicar ambos valores:
 - pos avanza dos posiciones (saltando de par a impar).
 - L aumenta en uno.
 - R disminuye en uno.

Este proceso simula exactamente el patrón con el que Policarpa organizó su secuencia, pero de forma invertida y mucho más eficiente.

2.3 Resultado

Probando el método con el mismo ejemplo, obtenemos nuevamente: 9 1 7 2

La respuesta es correcta, y su complejidad es $O(n)$, recorriendo la secuencia una sola vez.

Conclusión:

Ambos métodos permiten reconstruir correctamente la secuencia original escrita por Policarpa. Sin embargo:

El método cándido reproduce el razonamiento intuitivo pero es ineficiente para secuencias grandes.

El método óptimo, gracias al uso de two pointers, ofrece una solución clara, elegante y con eficiencia lineal.