Proyecto 3 – Análisis y Diseño de Algoritmos)

Sergio Orellana

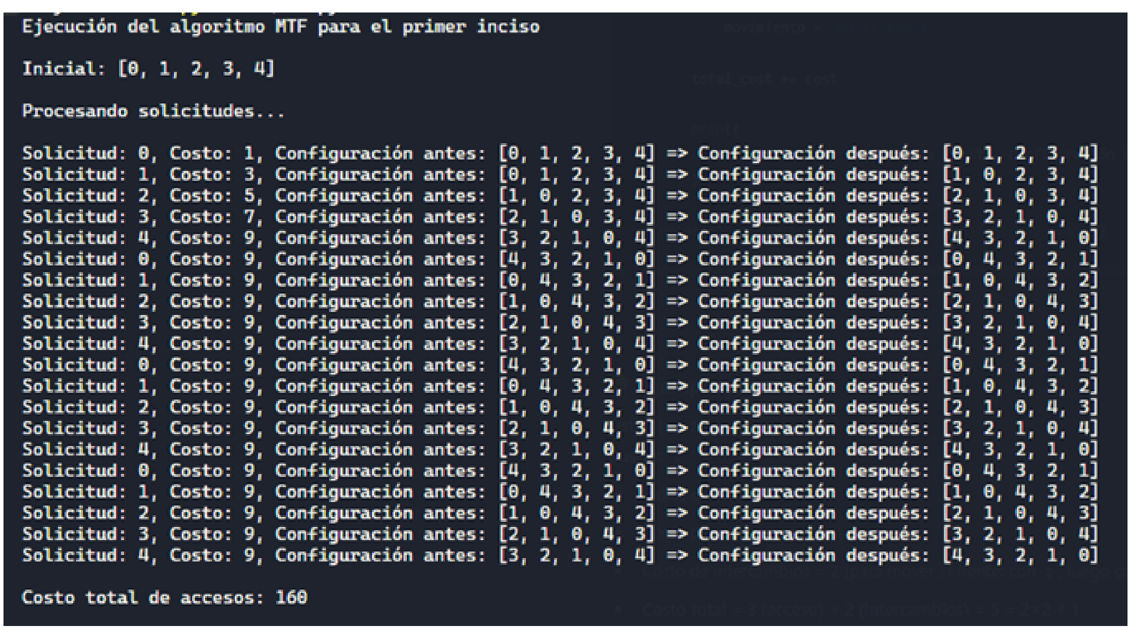
8 de abril de 2025

Inciso 1: Secuencia ordenada

Configuración inicial: [0, 1, 2, 3, 4]
Secuencia de solicitudes: [0, 1, 2, 3, 4] repetida 4 veces (20 solicitudes)

Explicación: En primer lugar, esta secuencia sigue un patrón repetitivo y completamente ordenado. Con el algoritmo MTF y bajo el modelo de costo actualizado, cada acceso implica un costo de búsqueda (*posición + 1*) y un costo de movimiento (*número de intercambios realizados*). Es decir, el costo total de cada acceso se calcula como **2 × posición + 1**. A medida que se procesan las solicitudes, la lista se reorganiza constantemente, generando un comportamiento cíclico que acumula un costo significativo.

Costo total: El costo total acumulado fue de **160**.



```
Ejecución del algoritmo MTF para el primer inciso

Inicial: [0, 1, 2, 3, 4]

Procesando solicitudes...

Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 1, Costo: 3, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [1, 0, 2, 3, 4]
Solicitud: 2, Costo: 5, Configuración antes: [1, 0, 2, 3, 4] => Configuración después: [2, 1, 0, 3, 4]
Solicitud: 3, Costo: 7, Configuración antes: [2, 1, 0, 3, 4] => Configuración después: [3, 2, 1, 0, 4]
Solicitud: 4, Costo: 9, Configuración antes: [3, 2, 1, 0, 4] => Configuración después: [4, 3, 2, 1, 0]
Solicitud: 0, Costo: 9, Configuración antes: [4, 3, 2, 1, 0] => Configuración después: [0, 4, 3, 2, 1]
Solicitud: 1, Costo: 9, Configuración antes: [0, 4, 3, 2, 1] => Configuración después: [1, 0, 4, 3, 2]
Solicitud: 2, Costo: 9, Configuración antes: [1, 0, 4, 3, 2] => Configuración después: [2, 1, 0, 4, 3]
Solicitud: 3, Costo: 9, Configuración antes: [2, 1, 0, 4, 3] => Configuración después: [3, 2, 1, 0, 4]
Solicitud: 4, Costo: 9, Configuración antes: [3, 2, 1, 0, 4] => Configuración después: [4, 3, 2, 1, 0]
Solicitud: 0, Costo: 9, Configuración antes: [4, 3, 2, 1, 0] => Configuración después: [0, 4, 3, 2, 1]
Solicitud: 1, Costo: 9, Configuración antes: [0, 4, 3, 2, 1] => Configuración después: [1, 0, 4, 3, 2]
Solicitud: 2, Costo: 9, Configuración antes: [1, 0, 4, 3, 2] => Configuración después: [2, 1, 0, 4, 3]
Solicitud: 3, Costo: 9, Configuración antes: [2, 1, 0, 4, 3] => Configuración después: [3, 2, 1, 0, 4]
Solicitud: 4, Costo: 9, Configuración antes: [3, 2, 1, 0, 4] => Configuración después: [4, 3, 2, 1, 0]
Solicitud: 0, Costo: 9, Configuración antes: [4, 3, 2, 1, 0] => Configuración después: [0, 4, 3, 2, 1]
Solicitud: 1, Costo: 9, Configuración antes: [0, 4, 3, 2, 1] => Configuración después: [1, 0, 4, 3, 2]
Solicitud: 2, Costo: 9, Configuración antes: [1, 0, 4, 3, 2] => Configuración después: [2, 1, 0, 4, 3]
Solicitud: 3, Costo: 9, Configuración antes: [2, 1, 0, 4, 3] => Configuración después: [3, 2, 1, 0, 4]
Solicitud: 4, Costo: 9, Configuración antes: [3, 2, 1, 0, 4] => Configuración después: [4, 3, 2, 1, 0]

Costo total de accesos: 160
```

Inciso 2: Secuencia alternante y repetida

Configuración inicial: [0, 1, 2, 3, 4]
Secuencia de solicitudes: [4, 3, 2, 1, 0, 1, 2, 3, 4, 3, 2, 1, 0, 1, 2, 3, 4]

Explicación: Esta secuencia genera una reorganización constante, ya que alterna elementos que en cada iteración se encuentran en distintas posiciones. Gracias al reordenamiento de MTF, los elementos más recientes tienden a estar más cerca del frente. Sin embargo, al sumar el costo de los intercambios por cada movimiento, se evidencia un aumento notable en el total.

Costo total: La suma total de los accesos en este caso fue de **117**.

```
Ejecución del algoritmo MTF para el segundo inciso

Inicial: [0, 1, 2, 3, 4]

Procesando solicitudes...

Solicitud: 4, Costo: 9, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [4, 0, 1, 2, 3]
Solicitud: 3, Costo: 9, Configuración antes: [4, 0, 1, 2, 3] => Configuración después: [3, 4, 0, 1, 2]
Solicitud: 2, Costo: 9, Configuración antes: [3, 4, 0, 1, 2] => Configuración después: [2, 3, 4, 0, 1]
Solicitud: 1, Costo: 9, Configuración antes: [2, 3, 4, 0, 1] => Configuración después: [1, 2, 3, 4, 0]
Solicitud: 0, Costo: 9, Configuración antes: [1, 2, 3, 4, 0] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 1, Costo: 3, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [1, 0, 2, 3, 4]
Solicitud: 2, Costo: 5, Configuración antes: [1, 0, 2, 3, 4] => Configuración después: [2, 1, 0, 3, 4]
Solicitud: 3, Costo: 7, Configuración antes: [2, 1, 0, 3, 4] => Configuración después: [3, 2, 1, 0, 4]
Solicitud: 4, Costo: 9, Configuración antes: [3, 2, 1, 0, 4] => Configuración después: [4, 3, 2, 1, 0]
Solicitud: 3, Costo: 3, Configuración antes: [4, 3, 2, 1, 0] => Configuración después: [3, 4, 2, 1, 0]
Solicitud: 2, Costo: 5, Configuración antes: [3, 4, 2, 1, 0] => Configuración después: [2, 3, 4, 1, 0]
Solicitud: 1, Costo: 7, Configuración antes: [2, 3, 4, 1, 0] => Configuración después: [1, 2, 3, 4, 0]
Solicitud: 0, Costo: 9, Configuración antes: [1, 2, 3, 4, 0] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 1, Costo: 3, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [1, 0, 2, 3, 4]
Solicitud: 2, Costo: 5, Configuración antes: [1, 0, 2, 3, 4] => Configuración después: [2, 1, 0, 3, 4]
Solicitud: 3, Costo: 7, Configuración antes: [2, 1, 0, 3, 4] => Configuración después: [3, 2, 1, 0, 4]
Solicitud: 4, Costo: 9, Configuración antes: [3, 2, 1, 0, 4] => Configuración después: [4, 3, 2, 1, 0]

Costo total de accesos: 117
```

Inciso 3: Mejor caso posible para MTF

Pregunta: *¿Para qué secuencia de 20 solicitudes se obtiene el mínimo costo total de acceso utilizando MTF para la configuración [0, 1, 2, 3, 4]?*

Respuesta: El mejor caso ocurre cuando se accede repetidamente al elemento que ya está al frente de la lista. En este caso, el elemento no necesita ser reordenado, por lo tanto no hay intercambios, y el costo por acceso es mínimo.

Ejemplo de secuencia:

`[0, 0, 0, ..., 0]` (20 veces)

Costo total: Cada acceso cuesta 1 (posición 0: $2\times0 + 1$), así que el costo total es: $1 \times 20 =$ **20**.

```
Ejecución del algoritmo MTF para el tercer inciso

Inicial: [0, 1, 2, 3, 4]

Procesando solicitudes...

Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [0, 1, 2, 3, 4]

Costo total de accesos: 20
```

Inciso 4: Peor caso posible para MTF

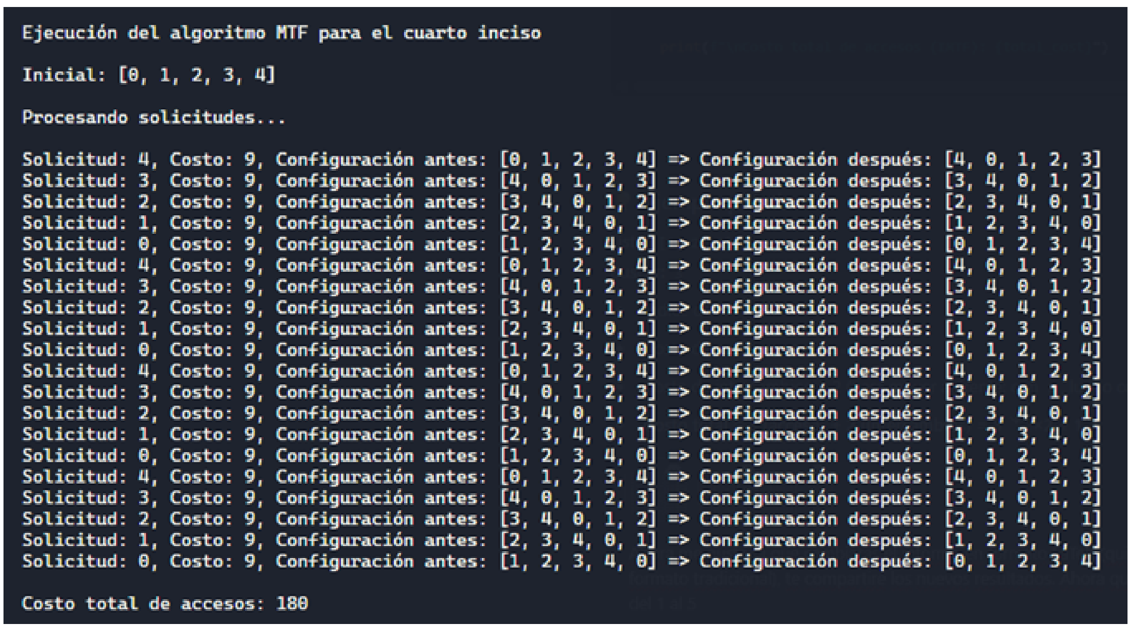
Pregunta: *¿Para qué secuencia de 20 solicitudes se obtiene el peor de los casos utilizando MTF para la configuración [0, 1, 2, 3, 4]?*

Respuesta: El peor escenario se da cuando cada solicitud accede al último elemento de la lista. Luego, al moverlo al frente, la siguiente solicitud apunta a otro que vuelve a estar al final. Este comportamiento provoca que cada acceso tenga el costo máximo.

Ejemplo:

`[4, 3, 2, 1, 0]` repetido 4 veces

Costo total: Cada elemento accedido está en la última posición (índice 4), así que el costo de cada acceso es $2 \times 4 + 1 = 9$. $9 \times 20 =$ **180**



```
Ejecución del algoritmo MTF para el cuarto inciso

Inicial: [0, 1, 2, 3, 4]

Procesando solicitudes...

Solicitud: 4, Costo: 9, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [4, 0, 1, 2, 3]
Solicitud: 3, Costo: 9, Configuración antes: [4, 0, 1, 2, 3] => Configuración después: [3, 4, 0, 1, 2]
Solicitud: 2, Costo: 9, Configuración antes: [3, 4, 0, 1, 2] => Configuración después: [2, 3, 4, 0, 1]
Solicitud: 1, Costo: 9, Configuración antes: [2, 3, 4, 0, 1] => Configuración después: [1, 2, 3, 4, 0]
Solicitud: 0, Costo: 9, Configuración antes: [1, 2, 3, 4, 0] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 4, Costo: 9, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [4, 0, 1, 2, 3]
Solicitud: 3, Costo: 9, Configuración antes: [4, 0, 1, 2, 3] => Configuración después: [3, 4, 0, 1, 2]
Solicitud: 2, Costo: 9, Configuración antes: [3, 4, 0, 1, 2] => Configuración después: [2, 3, 4, 0, 1]
Solicitud: 1, Costo: 9, Configuración antes: [2, 3, 4, 0, 1] => Configuración después: [1, 2, 3, 4, 0]
Solicitud: 0, Costo: 9, Configuración antes: [1, 2, 3, 4, 0] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 4, Costo: 9, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [4, 0, 1, 2, 3]
Solicitud: 3, Costo: 9, Configuración antes: [4, 0, 1, 2, 3] => Configuración después: [3, 4, 0, 1, 2]
Solicitud: 2, Costo: 9, Configuración antes: [3, 4, 0, 1, 2] => Configuración después: [2, 3, 4, 0, 1]
Solicitud: 1, Costo: 9, Configuración antes: [2, 3, 4, 0, 1] => Configuración después: [1, 2, 3, 4, 0]
Solicitud: 0, Costo: 9, Configuración antes: [1, 2, 3, 4, 0] => Configuración después: [0, 1, 2, 3, 4]
Solicitud: 4, Costo: 9, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [4, 0, 1, 2, 3]
Solicitud: 3, Costo: 9, Configuración antes: [4, 0, 1, 2, 3] => Configuración después: [3, 4, 0, 1, 2]
Solicitud: 2, Costo: 9, Configuración antes: [3, 4, 0, 1, 2] => Configuración después: [2, 3, 4, 0, 1]
Solicitud: 1, Costo: 9, Configuración antes: [2, 3, 4, 0, 1] => Configuración después: [1, 2, 3, 4, 0]
Solicitud: 0, Costo: 9, Configuración antes: [1, 2, 3, 4, 0] => Configuración después: [0, 1, 2, 3, 4]

Costo total de accesos: 180
```

Inciso 5: Repetición de un mismo elemento

Configuración inicial: [0, 1, 2, 3, 4]
Secuencias de solicitudes:

1. `[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]`
2. `[3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]`

Explicación: Cuando un mismo elemento se repite muchas veces, MTF lo lleva al frente tras la primera aparición. Como resultado, los accesos posteriores tienen un costo mínimo. Solo el primer acceso incurre en un costo alto debido a su posición original y el costo de moverlo al frente.

Costo total:

- Para el valor `2`: el primer acceso cuesta $2 \times 2 + 1 = 5$, y los 19 siguientes cuestan 1. Total $= 5 + 19 \times 1 =$ **24**
- Para el valor `3`: el primer acceso cuesta $2 \times 3 + 1 = 7$, y los 19 siguientes cuestan 1. Total $= 7 + 19 \times 1 =$ **26**

Observación: Este comportamiento demuestra cómo MTF se adapta favorablemente a secuencias con alta localidad temporal, beneficiando los accesos repetidos.

```
Ejecución del algoritmo MTF para el quinto inciso, secuencia de 2

Inicial: [0, 1, 2, 3, 4]

Procesando solicitudes...

Solicitud: 2, Costo: 5, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]
Solicitud: 2, Costo: 1, Configuración antes: [2, 0, 1, 3, 4] => Configuración después: [2, 0, 1, 3, 4]

Costo total de accesos: 24

Ejecución del algoritmo MTF para el sexto inciso, secuencia de 3

Inicial: [0, 1, 2, 3, 4]

Procesando solicitudes...

Solicitud: 3, Costo: 7, Configuración antes: [0, 1, 2, 3, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]
Solicitud: 3, Costo: 1, Configuración antes: [3, 0, 1, 2, 4] => Configuración después: [3, 0, 1, 2, 4]

Costo total de accesos: 26
```

Inciso 6: Implementación del algoritmo IMTF

Descripción: El algoritmo IMTF (Improved Move-To-Front) es una mejora sobre el clásico MTF, en el cual se aplica una estrategia de *mirada hacia adelante*. Específicamente, después de acceder a un elemento en la posición i de la lista, este solo se mueve al frente si aparece nuevamente dentro de los siguientes $i - 1$ elementos de la secuencia de solicitudes. Esta verificación evita realizar movimientos innecesarios.

Ventaja principal: Al evitar reordenamientos redundantes, IMTF logra reducir el costo total de acceso en secuencias donde los elementos no se repiten de forma inmediata. Este enfoque mejora el rendimiento en contextos con baja localidad temporal.

Secuencias evaluadas: En esta sección se evaluó el algoritmo únicamente con el mejor y el peor caso, definidos como:

- **Mejor caso:** `[0] * 20` — Acceso repetido al primer elemento.
- **Peor caso:** `[4, 3, 2, 1, 0] * 4` — Acceso constante a elementos al final de la lista.

Resultados obtenidos:

- En el **mejor caso**, ningún elemento fue movido al frente, y todos los accesos fueron al primer elemento. Esto produjo un costo total de **20**.
- En el **peor caso**, aunque los elementos accedidos se encontraban en posiciones profundas, el algoritmo evitó muchos movimientos innecesarios, logrando reducir el costo total a **60**, comparado con los **180** que habría generado el algoritmo MTF bajo el mismo escenario.

Análisis: En el mejor caso, como el elemento `0` ya estaba al frente, el algoritmo nunca lo movió. Cada acceso tuvo un costo de 1. En contraste, el algoritmo MTF habría movido el elemento al frente en cada acceso, acumulando innecesariamente operaciones aunque el costo total también habría sido 20.

En el peor caso, el MTF tradicional habría acumulado un costo de 180, ya que cada elemento en la posición 4 se mueve al frente constantemente. Sin embargo, IMTF solo realizó movimientos cuando se detectó que un elemento volvería a ser solicitado pronto, lo que permitió reducir el total a **60**. Esto evidencia un comportamiento adaptativo eficiente.

```
Ejecución del algoritmo IMTF para el mejor de los casos

Inicial: [0, 1, 2, 3, 4]

Procesando solicitudes con IMTF...

Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]

Costo total de accesos (IMTF): 20

Ejecución del algoritmo IMTF para el peor de los casos

Inicial: [0, 1, 2, 3, 4]

Procesando solicitudes con IMTF...

Solicitud: 4, Costo: 5, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 3, Costo: 4, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 2, Costo: 3, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 1, Costo: 2, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 4, Costo: 5, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 3, Costo: 4, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 2, Costo: 3, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 1, Costo: 2, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 4, Costo: 5, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 3, Costo: 4, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 4, Costo: 5, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 3, Costo: 4, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 2, Costo: 3, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 1, Costo: 2, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 4, Costo: 5, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 3, Costo: 4, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 2, Costo: 3, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 1, Costo: 2, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]
Solicitud: 0, Costo: 1, Configuración antes: [0, 1, 2, 3, 4] => No se mueve → Configuración después: [0, 1, 2, 3, 4]

Costo total de accesos (IMTF): 60
```

Conclusión: El algoritmo IMTF representa una mejora significativa sobre MTF en términos de costo total, particularmente cuando los elementos no se repiten inmediatamente. Su rendimiento más inteligente le permite evitar movimientos innecesarios y reducir el número de operaciones, resultando en un algoritmo más eficiente y práctico en contextos reales.

Repositorio del Proyecto

El código fuente, los algoritmos implementados y las ejecuciones correspondientes a cada inciso de este proyecto se encuentran disponibles en el siguiente repositorio de GitHub:

`https://github.com/SergioAle210/Proyecto3-ADA`

Enlace del video

El siguiente link contiene el video de la explicación del código y del proyecto:

`https://youtu.be/gMAUZsLFX8Y`

Referencias

1. Bentley, J. L., Sleator, D. D., Tarjan, R. E., & Wei, V. K. (1986). A locally adaptive data compression scheme. *Communications of the ACM*, 29(4), 320–330.

2. GeeksForGeeks. (n.d.). *Self Organizing List — Move to Front Method*. Recuperado de: `https://www.geeksforgeeks.org/self-organizing-list-move-front-method/`

3. Rao, M. S., & Raju, C. N. (2011). *Improved Move-To-Front (IMTF) Heuristic for Self-Organizing Lists*. arXiv preprint: `https://arxiv.org/abs/1105.0187`

4. Knuth, D. E. (1998). *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley.