Proyecto MTF

Universidad del Valle de Guatemala

Análisis y Diseño de Algoritmos

Douglas de León Molina - 18037

In [74]:
```
def mtf(_list:list, element:int):
    """ Move To Front algorithm. """

    cost = 0
    for i, el in enumerate(_list):
        if el == element:
            cost = i + 1
            sel = _list.pop(i)
            _list.insert(0, sel)

            break

    return _list, cost
```

Inciso A

In [75]:
```
config = [0, 1, 2, 3, 4]
requests = [0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4]

total_cost = 0
for req in requests:
    print("Initial config:", config)
    print("Request:", req)

    config, cost = mtf(config, req)

    print("Resulting list:", config)
    print("Cost =", cost, "\n")
    total_cost += cost

print("TOTAL COST =", total_cost)
```

```
Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 1
```

```
Resulting list: [1, 0, 2, 3, 4]
Cost = 2

Initial config: [1, 0, 2, 3, 4]
Request: 2
Resulting list: [2, 1, 0, 3, 4]
Cost = 3

Initial config: [2, 1, 0, 3, 4]
Request: 3
Resulting list: [3, 2, 1, 0, 4]
Cost = 4

Initial config: [3, 2, 1, 0, 4]
Request: 4
Resulting list: [4, 3, 2, 1, 0]
Cost = 5

Initial config: [4, 3, 2, 1, 0]
Request: 0
Resulting list: [0, 4, 3, 2, 1]
Cost = 5

Initial config: [0, 4, 3, 2, 1]
Request: 1
Resulting list: [1, 0, 4, 3, 2]
Cost = 5

Initial config: [1, 0, 4, 3, 2]
Request: 2
Resulting list: [2, 1, 0, 4, 3]
Cost = 5

Initial config: [2, 1, 0, 4, 3]
Request: 3
Resulting list: [3, 2, 1, 0, 4]
Cost = 5

Initial config: [3, 2, 1, 0, 4]
Request: 4
Resulting list: [4, 3, 2, 1, 0]
Cost = 5

Initial config: [4, 3, 2, 1, 0]
Request: 0
Resulting list: [0, 4, 3, 2, 1]
Cost = 5

Initial config: [0, 4, 3, 2, 1]
Request: 1
Resulting list: [1, 0, 4, 3, 2]
Cost = 5

Initial config: [1, 0, 4, 3, 2]
Request: 2
Resulting list: [2, 1, 0, 4, 3]
Cost = 5
```

```
Initial config: [2, 1, 0, 4, 3]
Request: 3
Resulting list: [3, 2, 1, 0, 4]
Cost = 5

Initial config: [3, 2, 1, 0, 4]
Request: 4
Resulting list: [4, 3, 2, 1, 0]
Cost = 5

Initial config: [4, 3, 2, 1, 0]
Request: 0
Resulting list: [0, 4, 3, 2, 1]
Cost = 5

Initial config: [0, 4, 3, 2, 1]
Request: 1
Resulting list: [1, 0, 4, 3, 2]
Cost = 5

Initial config: [1, 0, 4, 3, 2]
Request: 2
Resulting list: [2, 1, 0, 4, 3]
Cost = 5

Initial config: [2, 1, 0, 4, 3]
Request: 3
Resulting list: [3, 2, 1, 0, 4]
Cost = 5

Initial config: [3, 2, 1, 0, 4]
Request: 4
Resulting list: [4, 3, 2, 1, 0]
Cost = 5

TOTAL COST = 90
```

Inciso B

In [64]:

```
config = [0, 1, 2, 3, 4]
requests = [4, 3, 2, 1, 0, 1, 2, 3, 4, 3, 2, 1, 0, 1, 2, 3, 4]

total_cost = 0
for req in requests:
    print("Initial config:", config)
    print("Request:", req)

    config, cost = mtf(config, req)

    print("Resulting list:", config)
    print("Cost =", cost, "\n")
    total_cost += cost

print("TOTAL COST =", total_cost)
```

```
Initial config: [0, 1, 2, 3, 4]
Request: 4
Resulting list: [4, 0, 1, 2, 3]
Cost = 5

Initial config: [4, 0, 1, 2, 3]
Request: 3
Resulting list: [1, 4, 0, 2, 3]
Cost = 3

Initial config: [1, 4, 0, 2, 3]
Request: 2
Resulting list: [1, 4, 0, 2, 3]
Cost = 1

Initial config: [1, 4, 0, 2, 3]
Request: 1
Resulting list: [3, 1, 4, 0, 2]
Cost = 5

Initial config: [3, 1, 4, 0, 2]
Request: 0
Resulting list: [0, 3, 1, 4, 2]
Cost = 4

Initial config: [0, 3, 1, 4, 2]
Request: 1
Resulting list: [4, 0, 3, 1, 2]
Cost = 4

Initial config: [4, 0, 3, 1, 2]
Request: 2
Resulting list: [1, 4, 0, 3, 2]
Cost = 4

Initial config: [1, 4, 0, 3, 2]
Request: 3
Resulting list: [3, 1, 4, 0, 2]
```

```
Cost = 4

Initial config: [3, 1, 4, 0, 2]
Request: 4
Resulting list: [4, 3, 1, 0, 2]
Cost = 3

Initial config: [4, 3, 1, 0, 2]
Request: 3
Resulting list: [4, 3, 1, 0, 2]
Cost = 1

Initial config: [4, 3, 1, 0, 2]
Request: 2
Resulting list: [3, 4, 1, 0, 2]
Cost = 2

Initial config: [3, 4, 1, 0, 2]
Request: 1
Resulting list: [2, 3, 4, 1, 0]
Cost = 5

Initial config: [2, 3, 4, 1, 0]
Request: 0
Resulting list: [4, 2, 3, 1, 0]
Cost = 3

Initial config: [4, 2, 3, 1, 0]
Request: 1
Resulting list: [3, 4, 2, 1, 0]
Cost = 3

Initial config: [3, 4, 2, 1, 0]
Request: 2
Resulting list: [2, 3, 4, 1, 0]
Cost = 3

Initial config: [2, 3, 4, 1, 0]
Request: 3
Resulting list: [3, 2, 4, 1, 0]
Cost = 2

Initial config: [3, 2, 4, 1, 0]
Request: 4
Resulting list: [3, 2, 4, 1, 0]
Cost = 1

TOTAL COST = 53
```

Inciso C

¿Para qué secuencia de 20 solicitudes se obtiene el mínimo costo total de acceso utilizando el algoritmo MTF para la configuración 0, 1, 2, 3, 4? ¿Cuál sería ese costo total de acceso?

In [80]:

```
config = [0, 1, 2, 3, 4]
best_case = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

print("El mínimo costo estaría dado por la secuencia:", best_case)

total_cost = 0
for req in best_case:
    print("Initial config:", config)
    print("Request:", req)

    config, cost = mtf(config, req)

    print("Resulting list:", config)
    print("Cost =", cost, "\n")
    total_cost += cost

print("TOTAL COST =", total_cost)
```

```
El mínimo costo estaría dado por la secuencia: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1
```

```
Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
```

```
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

TOTAL COST = 20
```

Inciso D

¿Para qué secuencia de 20 solicitudes se obtiene el peor de los casos utilizando el algoritmo MTF para la configuración 0, 1, 2, 3, 4? ¿Cuál sería ese costo total de acceso?

In [78]:

```
config = [0, 1, 2, 3, 4]
worst_case = [4, 3, 2, 1, 0, 4, 3, 2, 1, 0, 4, 3, 2, 1, 0, 4, 3, 2, 1, 0]

print("El máximo costo estaría dado por la secuencia:", worst_case)

total_cost = 0
for req in worst_case:
    print("Initial config:", config)
    print("Request:", req)

    config, cost = mtf(config, req)

    print("Resulting list:", config)
    print("Cost =", cost, "\n")
    total_cost += cost

print("TOTAL COST =", total_cost)
```

```
El máximo costo estaría dado por la secuencia: [4, 3, 2, 1, 0, 4, 3, 2, 1, 0,
4, 3, 2, 1, 0, 4, 3, 2, 1, 0]
Initial config: [0, 1, 2, 3, 4]
Request: 4
Resulting list: [4, 0, 1, 2, 3]
Cost = 5

Initial config: [4, 0, 1, 2, 3]
Request: 3
Resulting list: [3, 4, 0, 1, 2]
Cost = 5

Initial config: [3, 4, 0, 1, 2]
Request: 2
Resulting list: [2, 3, 4, 0, 1]
Cost = 5

Initial config: [2, 3, 4, 0, 1]
```

```
Request: 1
Resulting list: [1, 2, 3, 4, 0]
Cost = 5

Initial config: [1, 2, 3, 4, 0]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 5

Initial config: [0, 1, 2, 3, 4]
Request: 4
Resulting list: [4, 0, 1, 2, 3]
Cost = 5

Initial config: [4, 0, 1, 2, 3]
Request: 3
Resulting list: [3, 4, 0, 1, 2]
Cost = 5

Initial config: [3, 4, 0, 1, 2]
Request: 2
Resulting list: [2, 3, 4, 0, 1]
Cost = 5

Initial config: [2, 3, 4, 0, 1]
Request: 1
Resulting list: [1, 2, 3, 4, 0]
Cost = 5

Initial config: [1, 2, 3, 4, 0]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 5

Initial config: [0, 1, 2, 3, 4]
Request: 4
Resulting list: [4, 0, 1, 2, 3]
Cost = 5

Initial config: [4, 0, 1, 2, 3]
Request: 3
Resulting list: [3, 4, 0, 1, 2]
Cost = 5

Initial config: [3, 4, 0, 1, 2]
Request: 2
Resulting list: [2, 3, 4, 0, 1]
Cost = 5

Initial config: [2, 3, 4, 0, 1]
Request: 1
Resulting list: [1, 2, 3, 4, 0]
Cost = 5

Initial config: [1, 2, 3, 4, 0]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
```

```
Cost = 5

Initial config: [0, 1, 2, 3, 4]
Request: 4
Resulting list: [4, 0, 1, 2, 3]
Cost = 5

Initial config: [4, 0, 1, 2, 3]
Request: 3
Resulting list: [3, 4, 0, 1, 2]
Cost = 5

Initial config: [3, 4, 0, 1, 2]
Request: 2
Resulting list: [2, 3, 4, 0, 1]
Cost = 5

Initial config: [2, 3, 4, 0, 1]
Request: 1
Resulting list: [1, 2, 3, 4, 0]
Cost = 5

Initial config: [1, 2, 3, 4, 0]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 5

TOTAL COST = 100
```

Inciso E

In [81]:

```
config = [0, 1, 2, 3, 4]
requests = [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

total_cost = 0
for req in requests:
    print("Initial config:", config)
    print("Request:", req)

    config, cost = mtf(config, req)

    print("Resulting list:", config)
    print("Cost =", cost, "\n")
    total_cost += cost

print("TOTAL COST =", total_cost)
```

```
Initial config: [0, 1, 2, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 3

Initial config: [2, 0, 1, 3, 4]
Request: 2
```

```
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1
```

```
Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

Initial config: [2, 0, 1, 3, 4]
Request: 2
Resulting list: [2, 0, 1, 3, 4]
Cost = 1

TOTAL COST = 22
```

In [82]:

```
config = [0, 1, 2, 3, 4]
requests = [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]

total_cost = 0
for req in requests:
    print("Initial config:", config)
    print("Request:", req)

    config, cost = mtf(config, req)

    print("Resulting list:", config)
    print("Cost =", cost, "\n")
    total_cost += cost

print("TOTAL COST =", total_cost)
```

```
Initial config: [0, 1, 2, 3, 4]
```

```
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 4

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
```

```
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

Initial config: [3, 0, 1, 2, 4]
Request: 3
Resulting list: [3, 0, 1, 2, 4]
Cost = 1

TOTAL COST = 23
```

¿Cuál sería la fórmula para calcular el costo de n solicitudes del mismo elemento si éste se encuentra inicialmente en la k-ésma posición de la lista de configuración?

- El costo es de n+k

Inciso F

In [90]:
```python
def imtf(_list:list, element:int, requests:list):
    """ Move To Front algorithm. """
    cost = 0
    for i, el in enumerate(_list):
        if el == element:
            cost = i + 1
            if element in requests[:i]:
                sel = _list.pop(i)
                _list.insert(0, sel)

            break

    return _list, cost
```

In [91]:
```python
config = [0, 1, 2, 3, 4]
best_case = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

print("Con IMTF el mínimo costo estaría dado por la secuencia:", best_case)

total_cost = 0
for idx in range(len(best_case)):
    req = best_case.pop(0)
    print("Initial config:", config)
    print("Request:", req)

    config, cost = imtf(config, req, best_case)

    print("Resulting list:", config)
    print("Cost =", cost, "\n")
    total_cost += cost

print("TOTAL COST =", total_cost)
```

```
Con IMTF el mínimo costo estaría dado por la secuencia: [0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
```

```
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1
```

```
Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

TOTAL COST = 20
```

In [95]:

```
config = [0, 1, 2, 3, 4]
worst_case = [4, 3, 2, 1, 0, 4, 3, 2, 1, 0, 4, 3, 2, 1, 0, 4, 3, 2, 1, 0]

print("Con IMTF el costo máximo estaría dado por la secuencia:", worst_case)

total_cost = 0
for i in range(len(worst_case)):
    req = worst_case.pop(0)
    print("Initial config:", config)
    print("Request:", req)

    config, cost = imtf(config, req, worst_case)

    print("Resulting list:", config)
    print("Cost =", cost, "\n")
    total_cost += cost

print("TOTAL COST =", total_cost)
```

```
Con IMTF el costo máximo estaría dado por la secuencia: [4, 3, 2, 1, 0, 4, 3,
2, 1, 0, 4, 3, 2, 1, 0, 4, 3, 2, 1, 0]
Initial config: [0, 1, 2, 3, 4]
Request: 4
Resulting list: [0, 1, 2, 3, 4]
Cost = 5

Initial config: [0, 1, 2, 3, 4]
```

```
Request: 3
Resulting list: [0, 1, 2, 3, 4]
Cost = 4

Initial config: [0, 1, 2, 3, 4]
Request: 2
Resulting list: [0, 1, 2, 3, 4]
Cost = 3

Initial config: [0, 1, 2, 3, 4]
Request: 1
Resulting list: [0, 1, 2, 3, 4]
Cost = 2

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 4
Resulting list: [0, 1, 2, 3, 4]
Cost = 5

Initial config: [0, 1, 2, 3, 4]
Request: 3
Resulting list: [0, 1, 2, 3, 4]
Cost = 4

Initial config: [0, 1, 2, 3, 4]
Request: 2
Resulting list: [0, 1, 2, 3, 4]
Cost = 3

Initial config: [0, 1, 2, 3, 4]
Request: 1
Resulting list: [0, 1, 2, 3, 4]
Cost = 2

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 4
Resulting list: [0, 1, 2, 3, 4]
Cost = 5

Initial config: [0, 1, 2, 3, 4]
Request: 3
Resulting list: [0, 1, 2, 3, 4]
Cost = 4

Initial config: [0, 1, 2, 3, 4]
Request: 2
Resulting list: [0, 1, 2, 3, 4]
```

```
Cost = 3

Initial config: [0, 1, 2, 3, 4]
Request: 1
Resulting list: [0, 1, 2, 3, 4]
Cost = 2

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

Initial config: [0, 1, 2, 3, 4]
Request: 4
Resulting list: [0, 1, 2, 3, 4]
Cost = 5

Initial config: [0, 1, 2, 3, 4]
Request: 3
Resulting list: [0, 1, 2, 3, 4]
Cost = 4

Initial config: [0, 1, 2, 3, 4]
Request: 2
Resulting list: [0, 1, 2, 3, 4]
Cost = 3

Initial config: [0, 1, 2, 3, 4]
Request: 1
Resulting list: [0, 1, 2, 3, 4]
Cost = 2

Initial config: [0, 1, 2, 3, 4]
Request: 0
Resulting list: [0, 1, 2, 3, 4]
Cost = 1

TOTAL COST = 60
```

Inciso G

¿Cuál es la diferencia entre un algoritmo online y uno offline? ¿Cambiarían en algo su desempeño o su comportamiento MTF e IMTF si se usaran como algoritmos online (considere el efecto de diferentes secuencias de solicitudes)? Investigue y describa al menos un algoritmo adicional que sea online y que sirva para atender una secuencia de solicitudes de accesos.

- La diferencia entre un algoritmo online y uno offline es que uno online debe de procesar su input paso por paso de forma secuencial, mientras que uno offline tiene todo el input al momento de iniciar la ejecución.
- El algoritmo MTF no cambiaría, puesto que su comportamiento no depende necesariamente de las solicitudes siguientes sino que depende solamente de la configuración y la solicitud actual. Sin embargo, IMTF sí puede ser afectado puesto que depende de las siguientes i-1 solicitudes y en un momento dado es posible que no tenga acceso a todas esas solicitudes siguientes y no se pueda llegar a una decisión de si mover o no el elemento.
- El algoritmo de Insertion Sort es un algoritmo online, puesto que ordena los valores mientras se está ejecutando y no requiere de ninguna información previa del input.

In [ ]: