

①

```

Consultas.comenzar()
while (!consultas.fin()) {
    consulta = consultas.proximo();
    for (i = consulta.desde; i <= consulta.hasta; i++)
        cuenta[i] = cuenta[i] + consulta.valor;
}

```

← Algoritmo procesarMovimientos

Nº Cuentas (y consultas)	procesarMovimientos	procesarMovimientosOptimizado
1000	0,032 seg	0,013 seg
10000	0,437 seg	0,004 seg
25000	3,743 seg	0,012 seg
50000	13,227 seg	0,034 seg
100000	52,131 seg	0,018 seg

b) procesarMovimientos es eficiente porque es del orden de  $N^2$ , es decir crece siguiendo una función cuadrática

$$T(m) = \underbrace{1(m+1) + m \cdot \left( 1 + \underbrace{(3m+2) + m \cdot 2}_{\text{for}} \right)}_{\text{while}}$$

$$= m+1 + m + 3m^2 + 2m + 2m^2$$

$$= 5m^2 + 4m + 1 \rightarrow O(m^2)$$

c) procesarMovimientosOptimizado

$$T(m) = T(\text{for}_1) + T(\text{for}_2)$$

$$\text{for}_1 = (3m+2) + (m \cdot \text{cte})$$

$$\text{for}_2 = (3m+2) + m \left( \underbrace{3}_{\text{if}} + 2 \right)$$

entonces  $T(m) = 3m+2 + m \cdot \text{cte} + 3m+2 + 3m+2m \rightarrow O(m)$

Rta. este algoritmo, al ser  $O(m)$  es más eficiente que el primer algoritmo que es de  $O(m^2)$