

Ejercicio 2

Manuel AlSorda Torres
74017527-P

void ejercicio2 (vector<int> &v) {

int j, i=1, n=v.size();

if (n>1) do { // Mientras haya más de un elemento en v

int x=v[i]; // x=Valor del primer elemento de v

for (j=i; j>0; v[j]=v[j-1]; j--) {

v[j]=x;

v[j]=x;

i++;

} while (i<n);

Talla del problema v.size()

Tiempo

v=[1 2 3 4]

| Itera | j | i | n | x | v | [i] |
|-------|---|---|---|---|---|-----|
| 1° | 1 | 1 | 4 | 1 | | |
| 2° | 2 | 2 | 4 | 3 | | |
| 3° | 3 | 3 | 4 | 4 | | |
| 4 | | 4 | 4 | | | |

→ [1 2 2 3]

↓
i=n=stop

El código ordena vectores de menor a mayor
No presenta mejor o peor caso ya que
siempre recorre el vector completo.

Manuel Alberda Torres
74017527-P

Mejor Caso = Vector ordenado

$$C_i(n) = 1 + N \cdot (1) = N \\ = N \cdot \Omega(n)$$

1 2 3 4

Peor Caso = Vector ordenado inverso

$$C_s(n) = 1 + N \cdot \left(1 + 3 + \sum_{j=0}^{j=i} 1 \right) \\ = 1 + N \cdot (N) \\ = N^2 = O(N^2)$$

4 3 2 1
3 4 2 1
3 2 4 1
3 2 1 4
2 3 1 4
2 1 3 4
1 2 3 4

Ejercicio 3

Manuel Alberda Torres
74017527P

```
Unsigned ej3 (unsigned n) {  
    unsigned i = n, k = 0;  
    while (i > 0) {  
        unsigned j = i;  
        do {  
            j = j * 2;  
            k = k + 1;  
        } while (j <= n);  
        i = i / 2;  
    }  
}
```

Talla del problema = \sqrt{n} .

El problema no presenta mejor o peor caso.

$$C(n) = \frac{n}{2} \cdot \log n = \frac{n \log n}{2}$$