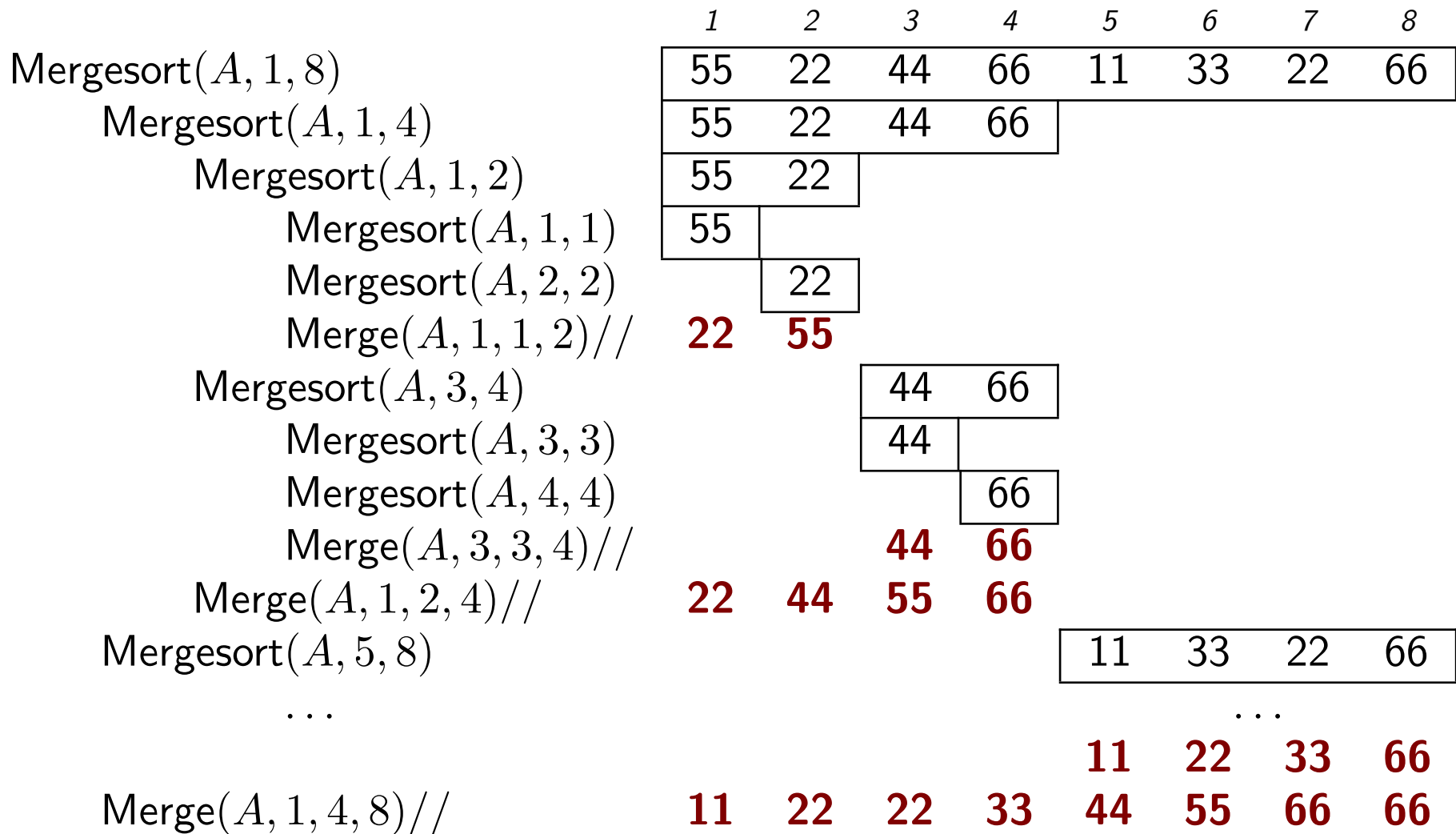


# Mergesort

## Mergesort: Traza



**Ejercicio. Traza.** Realiza una traza de Mergesort con  $A = \{ 3, 41, 52, 26, 38, 57, 9, 49 \}$ .

# Mergesort: análisis

Sea  $T(n)$  el tiempo de ejecución de mergesort. Si  $n = 1$ , el tiempo es constante. Si  $n > 1$  (supondremos por simplicidad que  $n$  es una potencia de 2):

- **Dividir:** Calcula el índice medio del vector:  $\Theta(1)$ .
- **Vencer:** Resuelve recursivamente los dos subproblemas, cada uno de ellos de talla  $n/2$ :  $2T(n/2)$ .
- **Combinar:** merge (fusión) de un vector de  $n$  elementos:  $\Theta(n)$ .

$$T(n) = \begin{cases} c_1, & \text{si } n \leq 1; \\ 2T(n/2) + c_2n + c_3, & \text{si } n > 1 \end{cases}$$

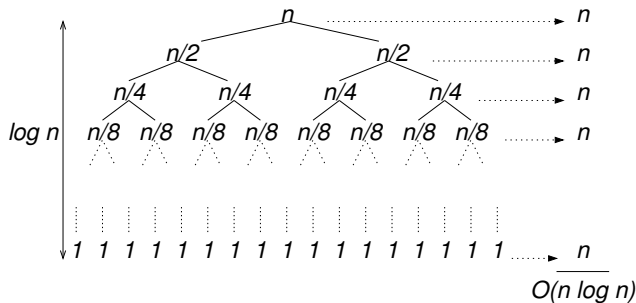
# Mergesort: análisis

$$\begin{aligned}
 T(n) &= 2T(n/2) + c_2n + c_3 \\
 &= 2(2T(n/2^2) + c_2n/2 + c_3) + c_2n + c_3 = \\
 &= 2^2T(n/2^2) + 2c_2n + c_3(2 + 1) \\
 &= 2^2(2T(n/2^3) + c_2n/2^2 + c_3) + 2c_2n + c_3(2 + 1) = \\
 &= 2^3T(n/2^3) + 3c_2n + c_3(2^2 + 2 + 1) \\
 &\vdots \quad \{i \text{ iteraciones}\} \\
 &= 2^i T(n/2^i) + ic_2n + c_3(2^{i-1} + \dots + 2^2 + 2 + 1)
 \end{aligned}$$

Es decir:

$$\begin{aligned}
 T(n) &= \{n/2^i = 1, i = \log n\}; \sum_{j=0}^{i-1} 2^j = 2^i - 1 \\
 &= nc_1 + c_2n \log n + c_3(n - 1) \in \Theta(n \log n)
 \end{aligned}$$

# División equilibrada de subproblemas



**Figura:** División equilibrada de subproblemas

$$T(n) = \begin{cases} c_1, & \text{si } n \leq 1; \\ 2T(n/2) + c_2n + c_3, & \text{si } n > 1 \end{cases}$$

# Mergesort con división no equilibrada

```
private static <T extends Comparable<T>>
void mergeSortBad(T v[], int izq, int der) {
    if (izq < der){
        mergeSort(v, izq, izq);
        mergeSort(v, izq+1, der);
        merge(v, izq, izq+1, der);
    }
}
```

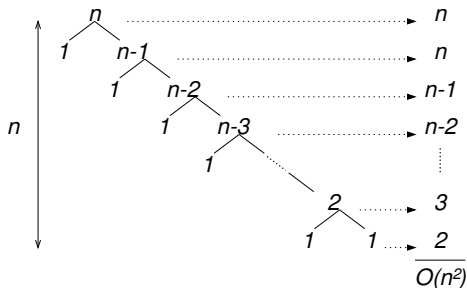


Figura: División no equilibrada

# Mergesort con división no equilibrada

Ecuaciones de recurrencia:

$$T(n) = \begin{cases} c_1, & \text{si } n \leq 1; \\ T(1) + T(n-1) + c_2n + c_3, & \text{si } n > 1 \end{cases}$$

$$T(n) = \begin{cases} c_1, & \text{si } n \leq 1; \\ T(n-1) + c_2n + c_4, & \text{si } n > 1 \end{cases}$$

Coste:  $\Theta(n^2)$