Estructuras de datos y Algoritmos II

UNIDAD 1: Algoritmos de ordenamiento

Profesor: Ing. Jonathan Roberto Torres Castillo

Ingenieria en Computación

Universidad Nacional Autónoma de México

Segunda sesión

1. Introducción MergeSort.
2. Descripción del Algoritmo Merge Sort.
3. Principio de Dividir
4. Principio de ordenar
5. Principio de Mezclar o intercalar.

Algoritmos de ordenamiento

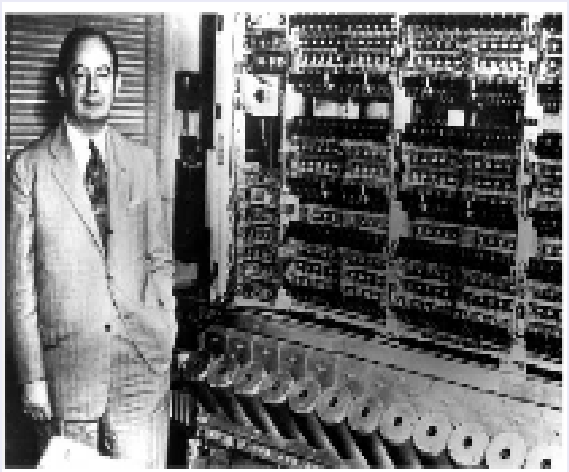
Ordenar una estructura de datos consiste en reacomodar sus elementos de manera tal que queden ordenados de acuerdo a un atributo clave. Los algoritmos de ordenamiento resultan un tema de interés por varios motivos:

1. Son importantes en diversas aplicaciones, en particular en el área de Bases de Datos, en donde los requirimientos de eficiencia hacen del ordenamiento un tema crítico.
2. Existen muchísimos métodos para resolver el mismo problema y por lo tanto es un tema interesante para introducir nociones de tiempo de ejecución y eficiencia.
3. Permiten ilustrar temas importantes de Resolución de Problemas.

Algoritmos de ordenamiento: MergeSort

Algoritmo de ordenación 'Divide y vencerás': Descripción y características

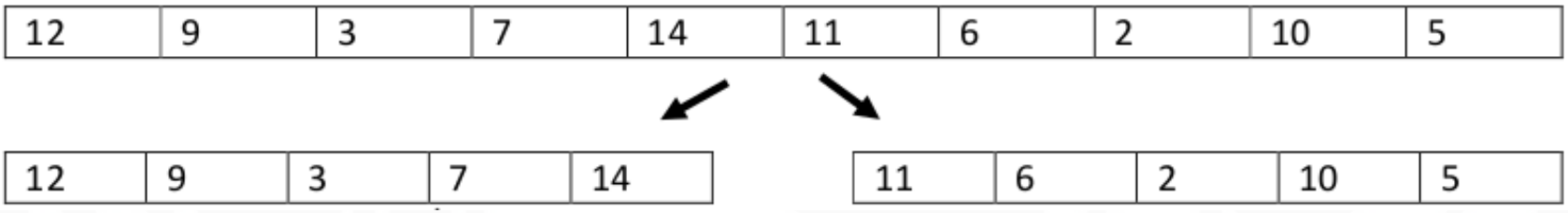
El algoritmo de ordenamiento por mezcla (*Merge Sort* en inglés) es un algoritmo de ordenamiento externo estable basado en la técnica divide y vencerás, también es llamado de intercalación o combinación, ya que intercala dos estructuras previamente ordenadas.



Fue desarrollado en 1945 por *Jonh Von Neuman*

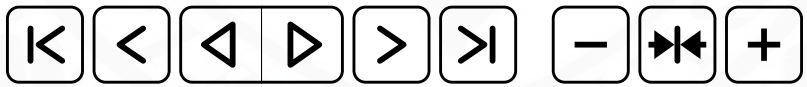
MergeSort: Dividir y ordenar

1. **'Divide y vencerás'** plantea que un problema puede ser dividido en varios **subproblemas** y una vez resueltos estos se puede proceder a **unir las soluciones** para formar la solución del problema general.



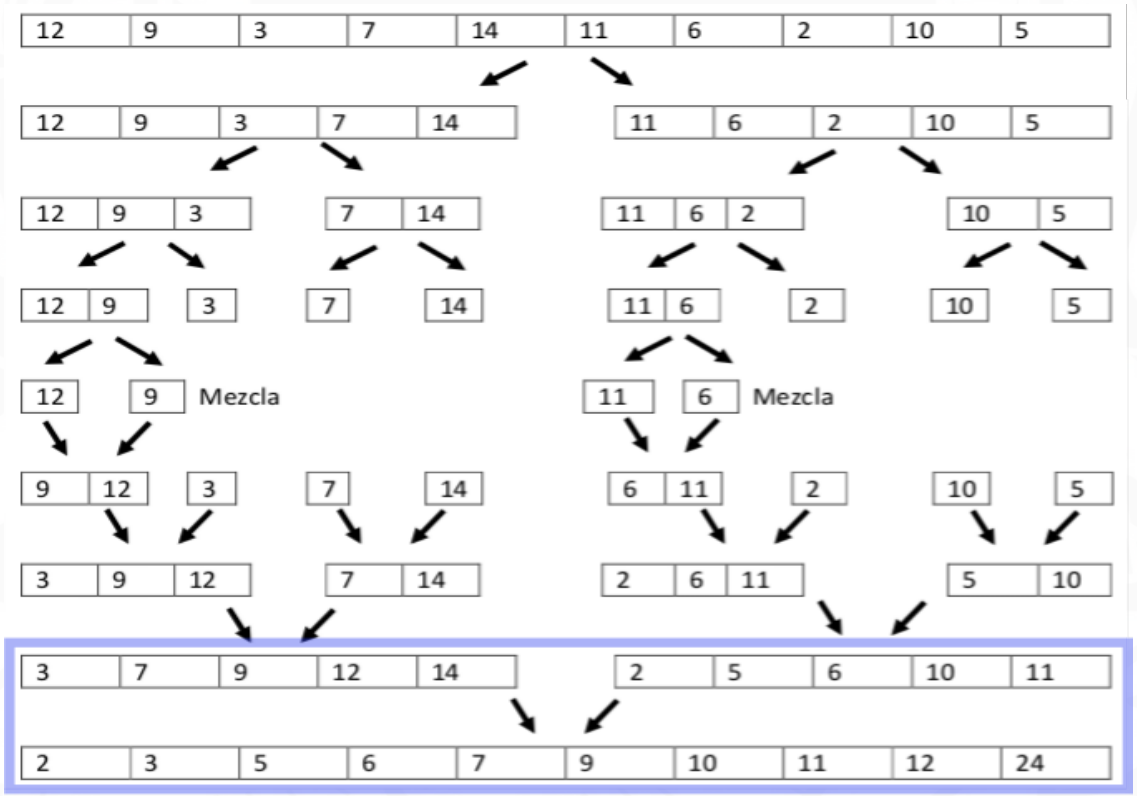
La solución de los **subproblemas** más pequeños se realiza de la misma manera: es decir, se van resolviendo problemas cada vez más pequeños (hasta llegar a un caso base: **problema no divisible** con solución **trivial**).

6 5 3 1 8 7 2 4



MergeSort

Cada recursión toma un arreglo de elementos desordenados. Se divide en dos mitades, se aplica la recursión en cada una de estas y ya que al finalizar estas recursiones tenemos las dos mitades ordenadas se **intercalan ambas** para obtener el arreglo ordenado.

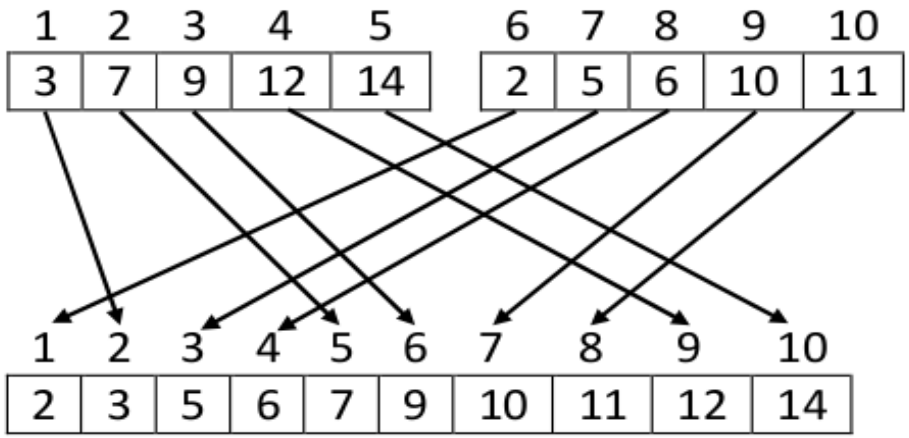


INTERCALAR

MergeSort: Mezclar-intercalar

Intercalar (Merge)

Es la operación que le da el nombre a este algoritmo. La intercalación toma dos secuencias (arreglos) de elementos y a partir de estas construye una tercera secuencia que contiene todos los elementos de estas en orden[1] [2] [3].



MergeSort: Algoritmo

El algoritmo de ordenamiento por mezcla o intercalación de un arreglo `A` consta de dos partes como se mostró anteriormente, el de división y ordenación (`MergeSort(A,p,r)`) y el de intercalación o mezcla (`Merge(A,p,q,r)`):

`MergeSort(A,p,r)`

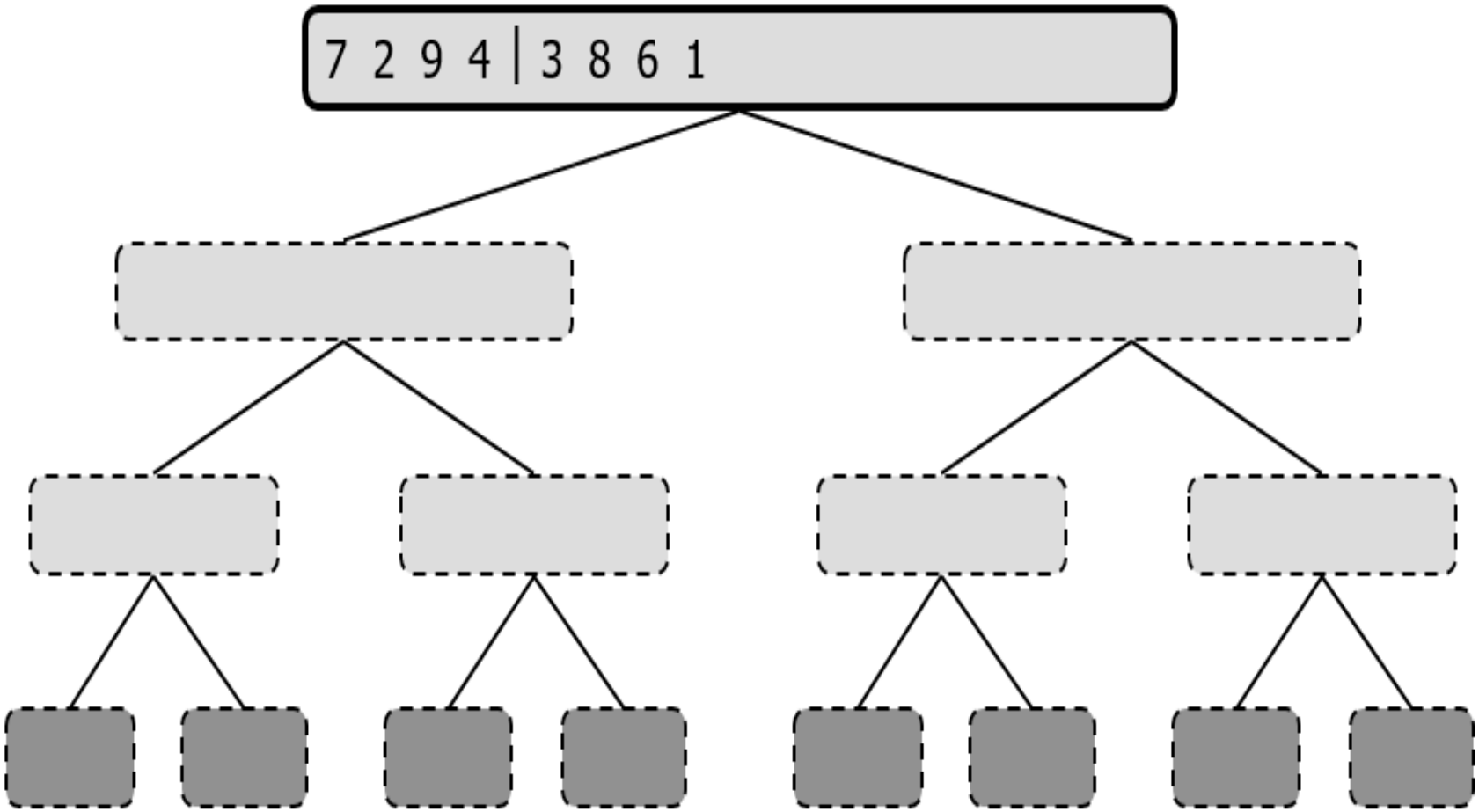
```
MergeSort(A,p,r)
Inicio
 Si p<r entonces
   q=(p+r)/2
   MergeSort(A, p , q)
   MergeSort(A, q+1 , r)
   Merge(A,p,q,r)
Fin
```

MergeSort: Algoritmo

```
Merge(A,p,q,r)
Inicio
Formar subarray Izq[0...q-p] y subarray Der[0...r-q]
Copiar de A[p...q] a Izq[0...q-p] y A[q+1...r]
... a Der[0...r-q-1]
i=0, j=0
Para k=p hasta r
    Si (j>=r-q) o (i<q-p+1 y Izq[i]<Der[j]) entonces
        A[k]=Izq[k]
        i=i+1
    En otro caso
        A[k]=Der[j]
        j=j+1
    Fin Si
Fin Para
Fin
```

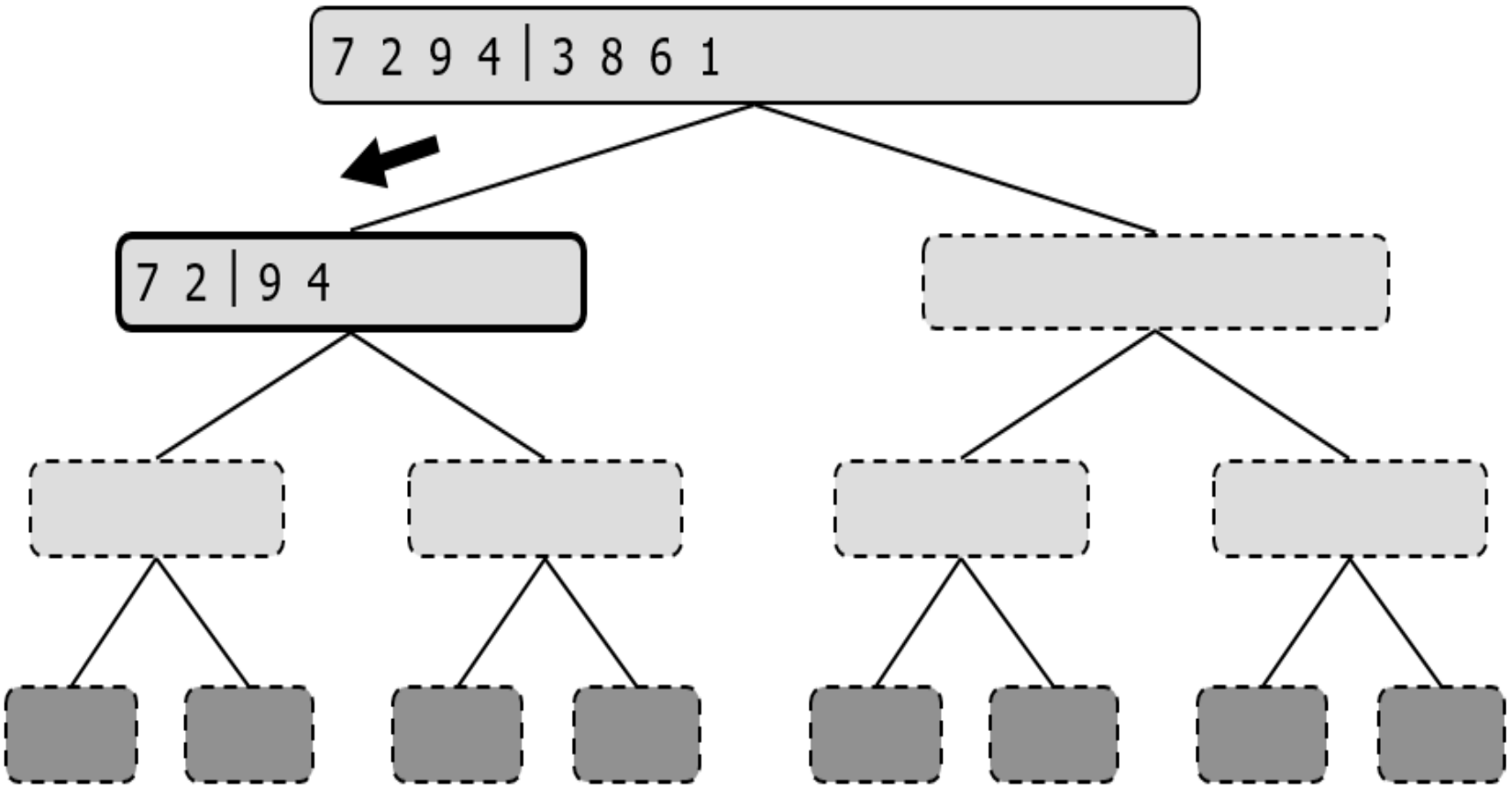
MergeSort: Ejemplo

Partición

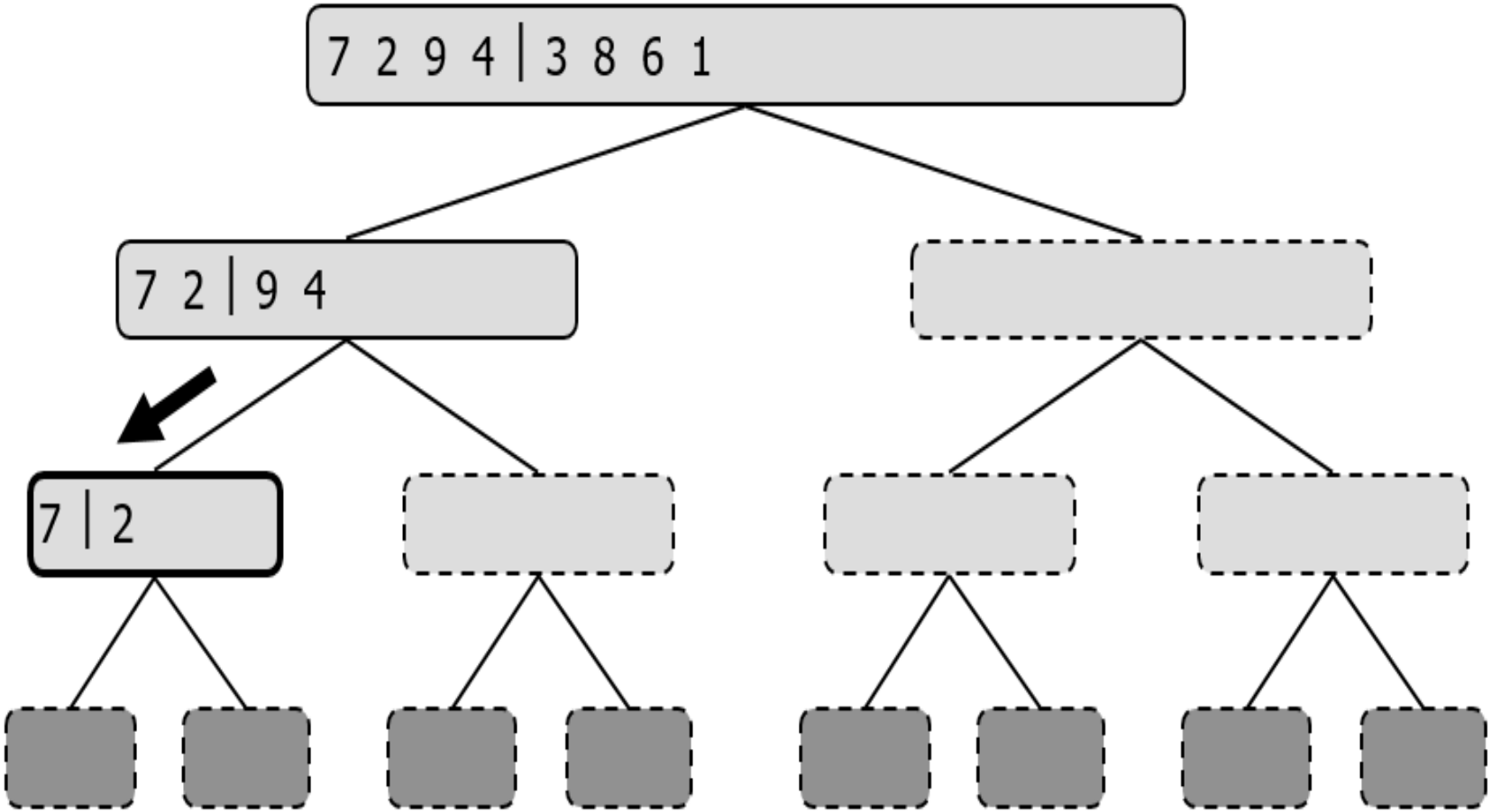


MergeSort: Ejemplo

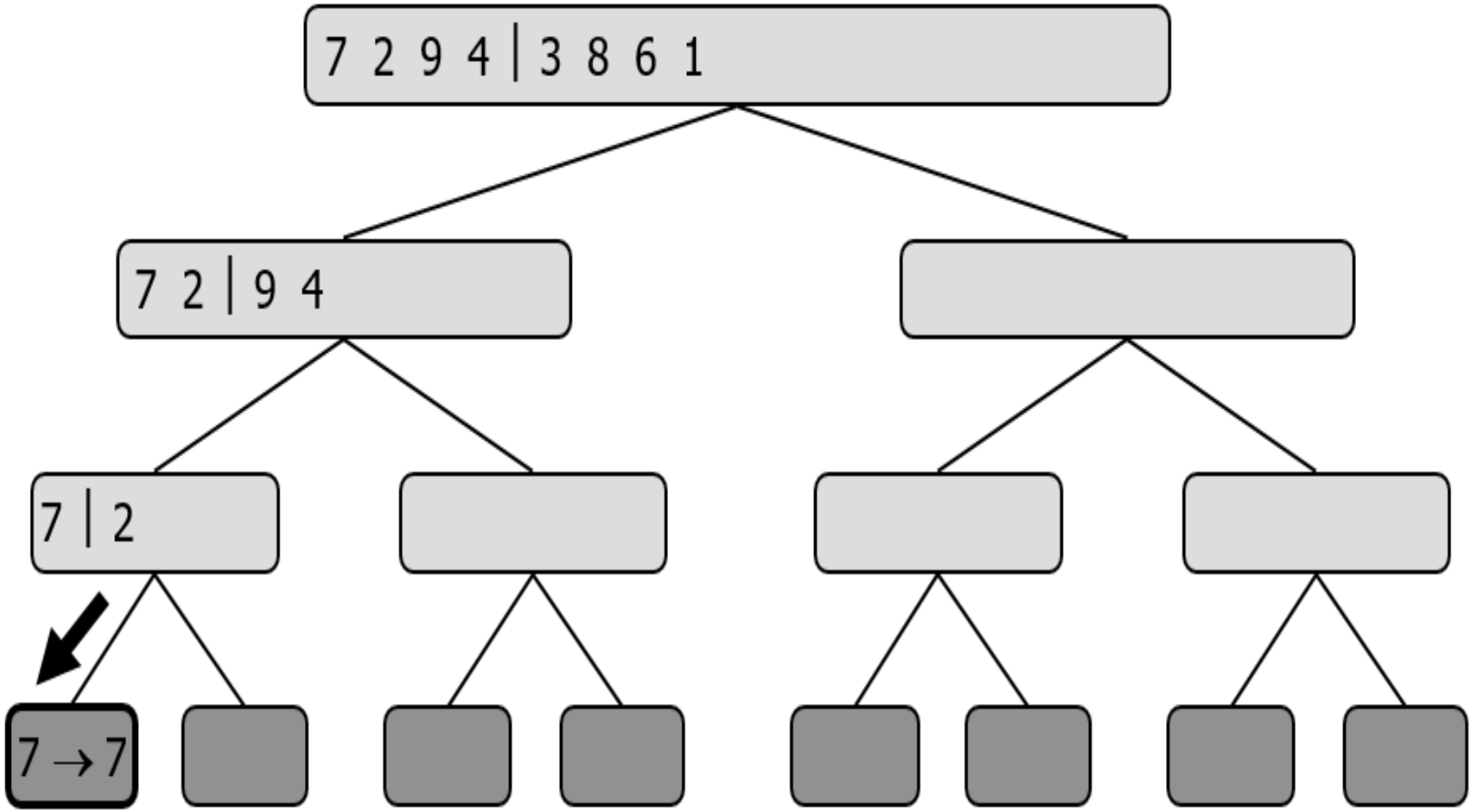
Llamada recursiva 1:



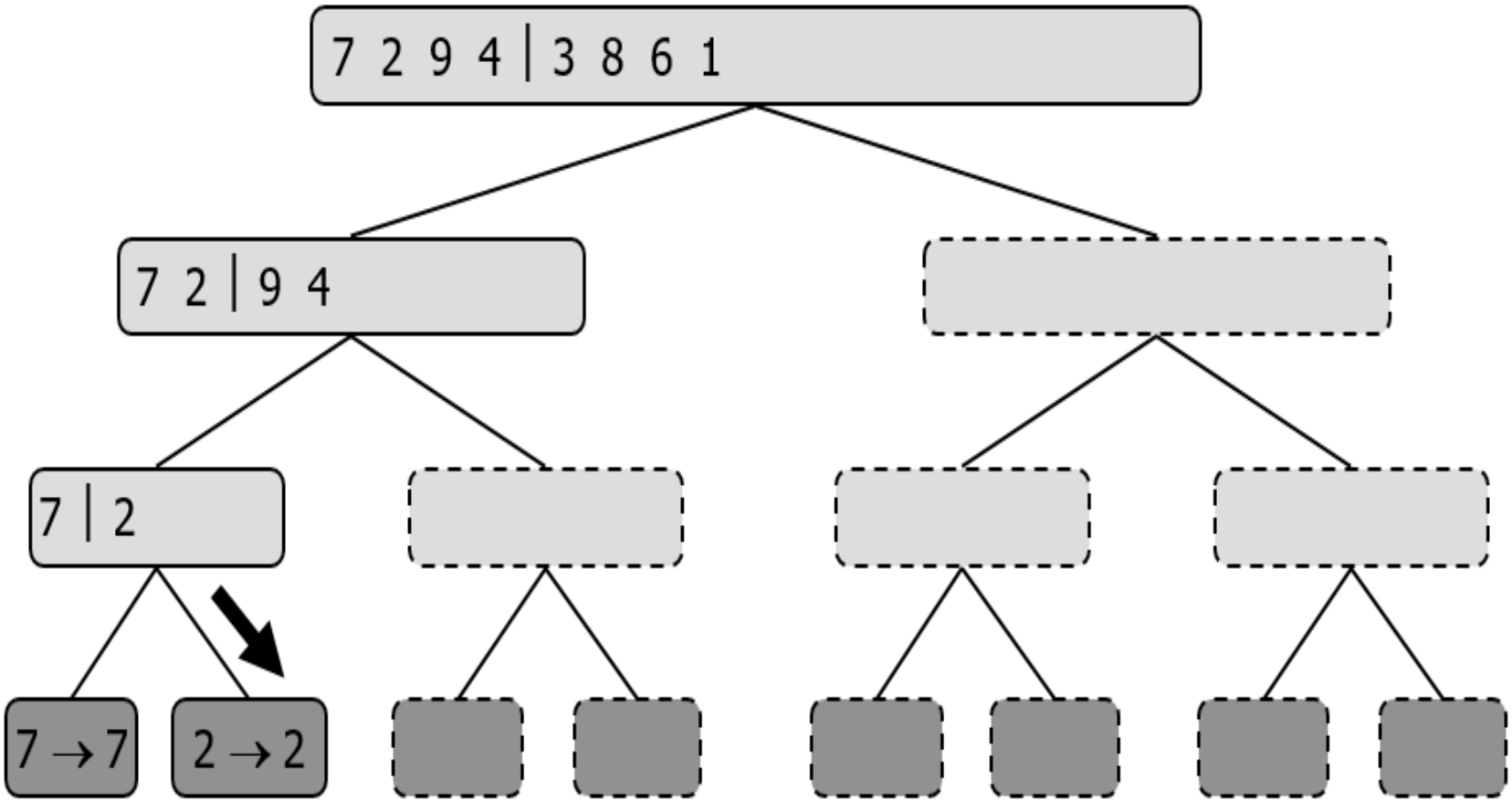
Llamada recursiva 2:



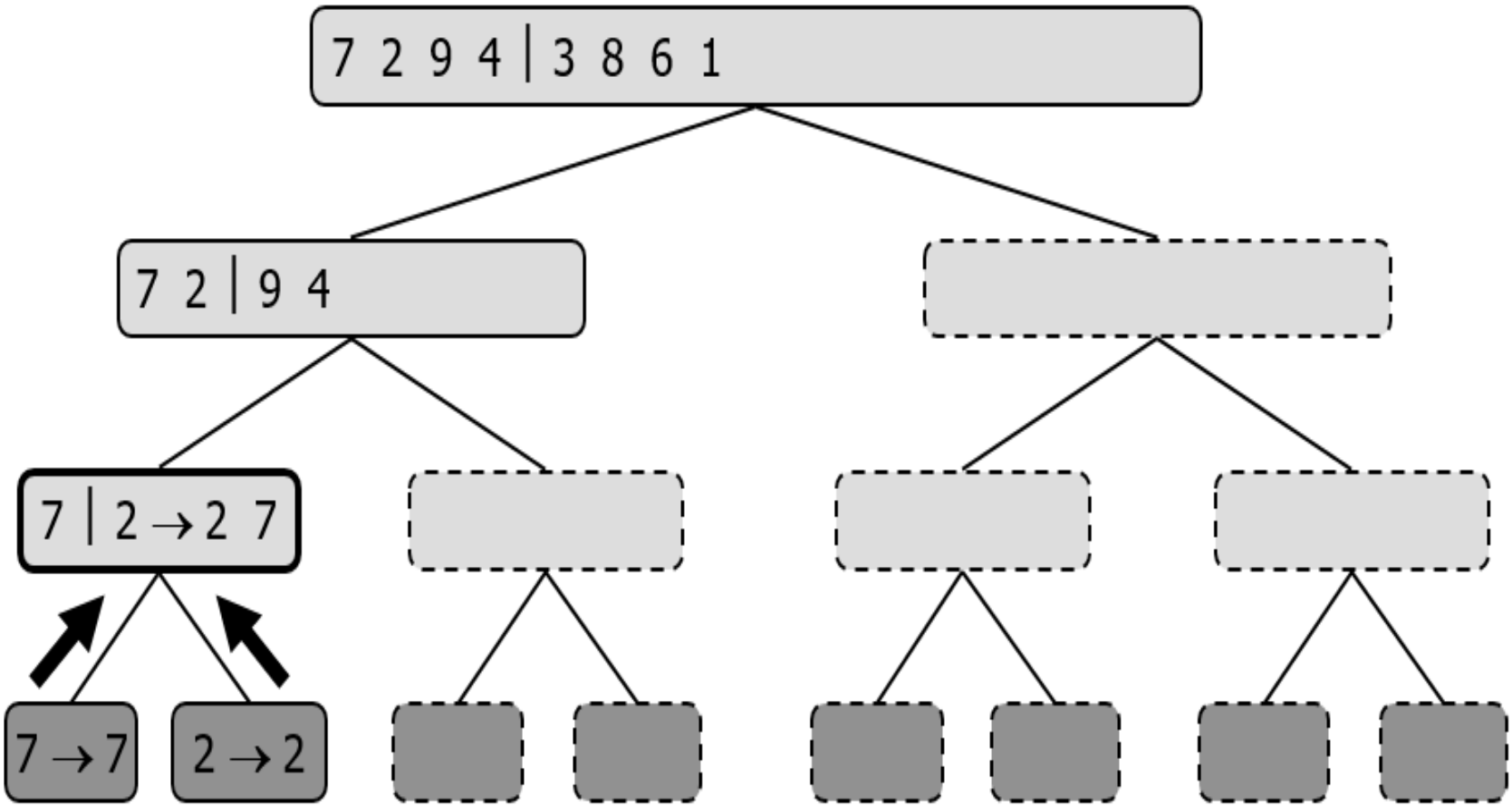
Llamada recursiva 3:



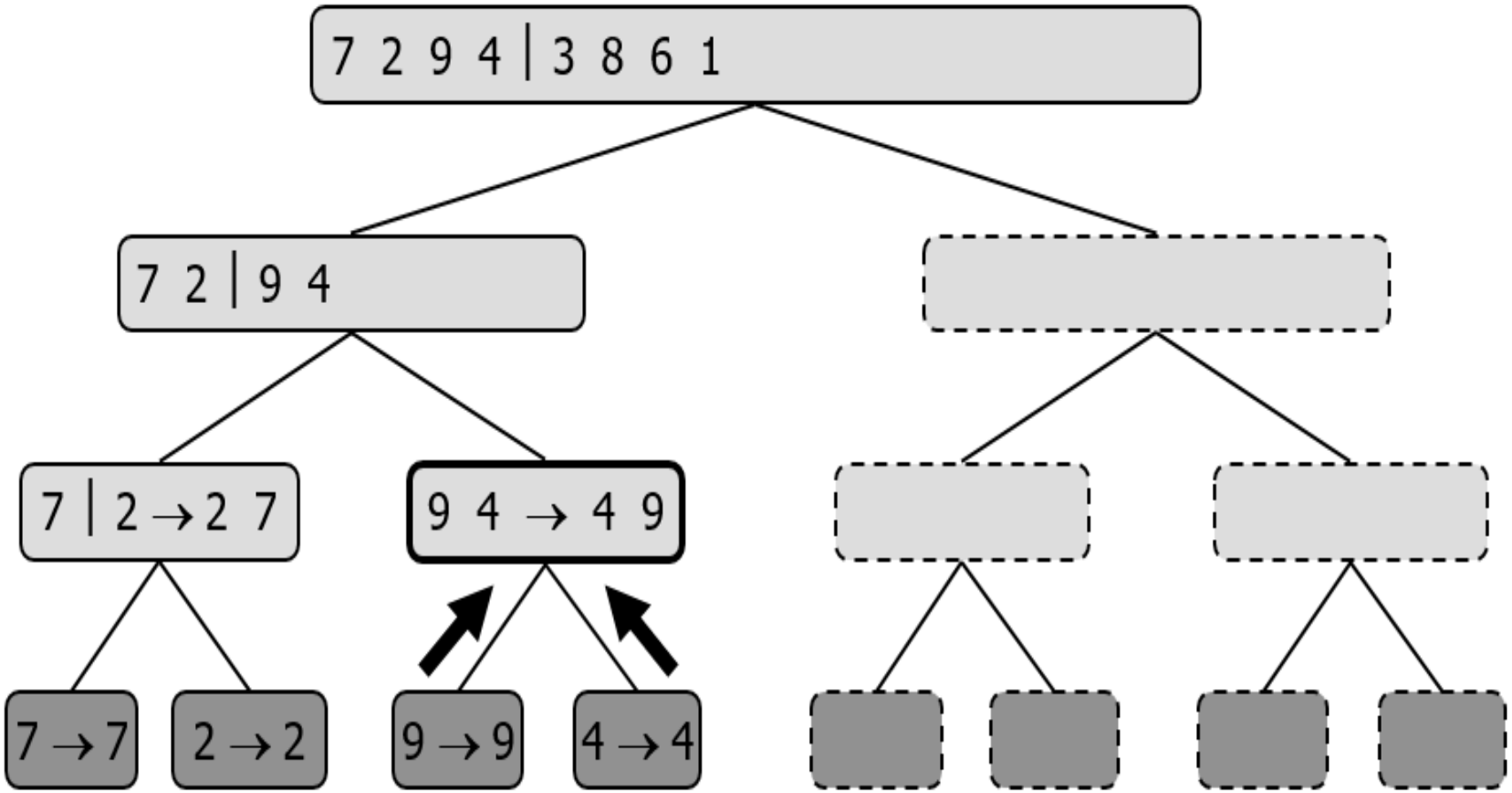
Llamada recursiva 4:



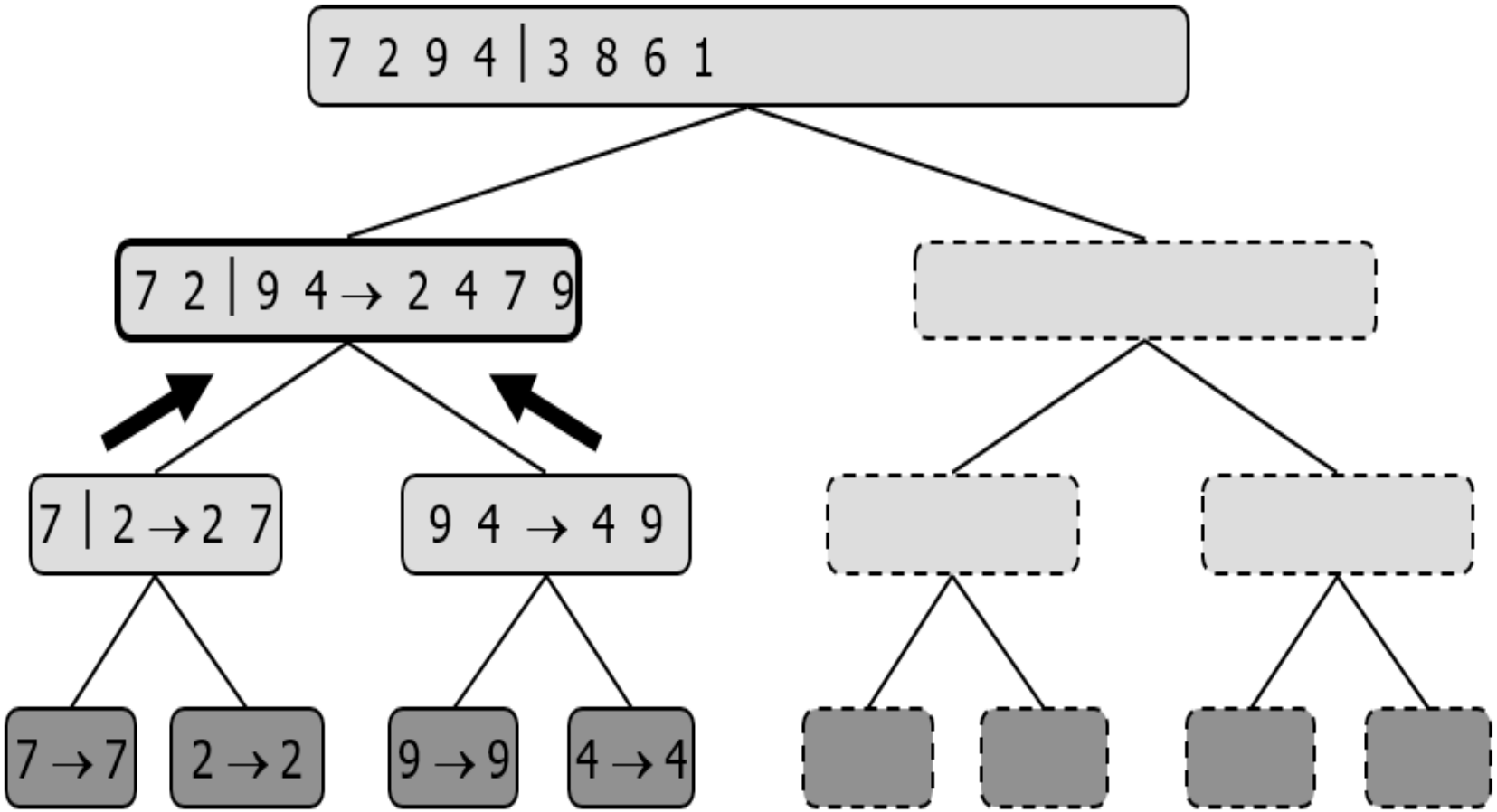
Mezcla (Merge)



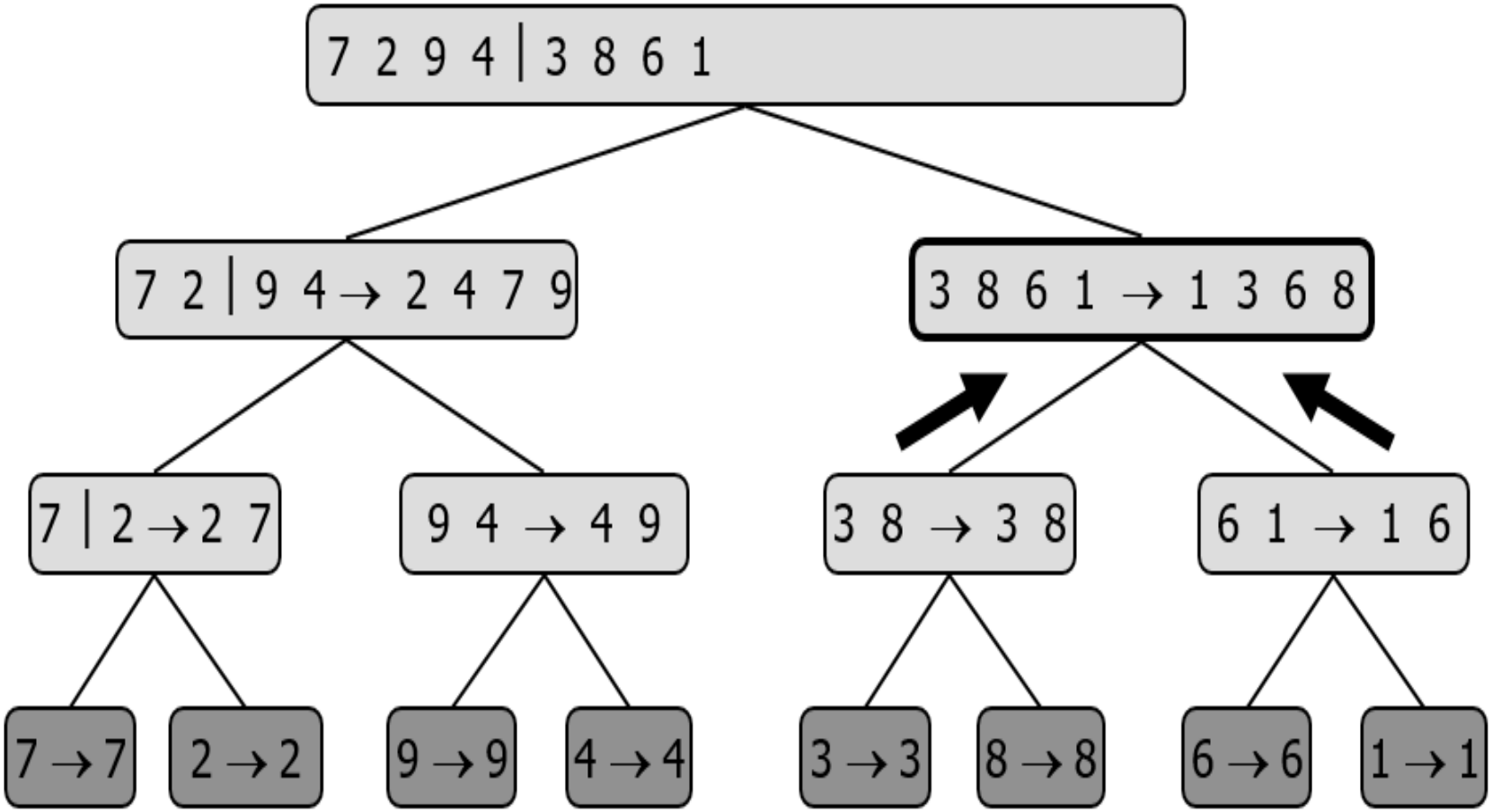
Llamada recursiva 5:



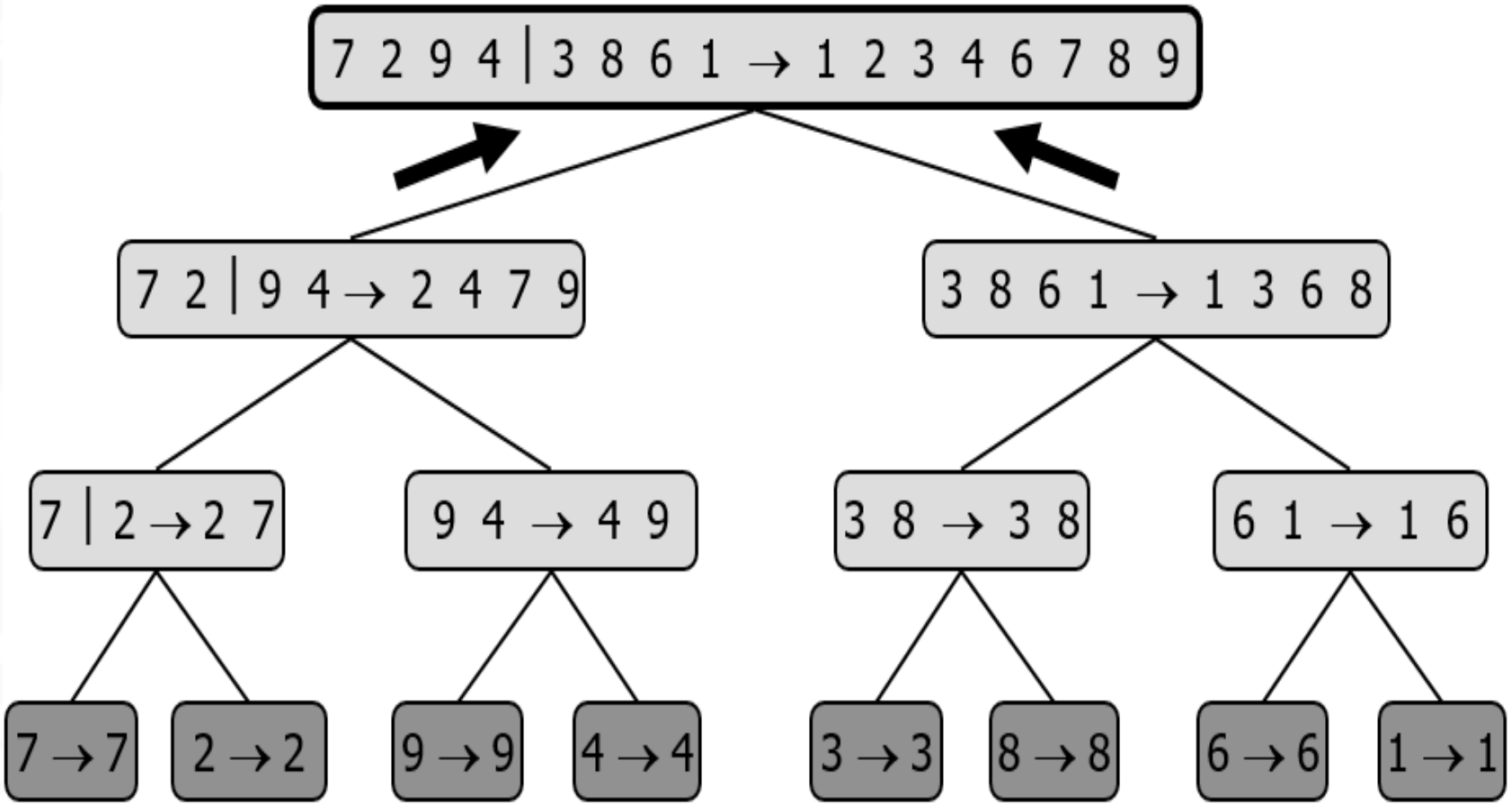
Mezcla (Merge)



Llamada recursiva 6:

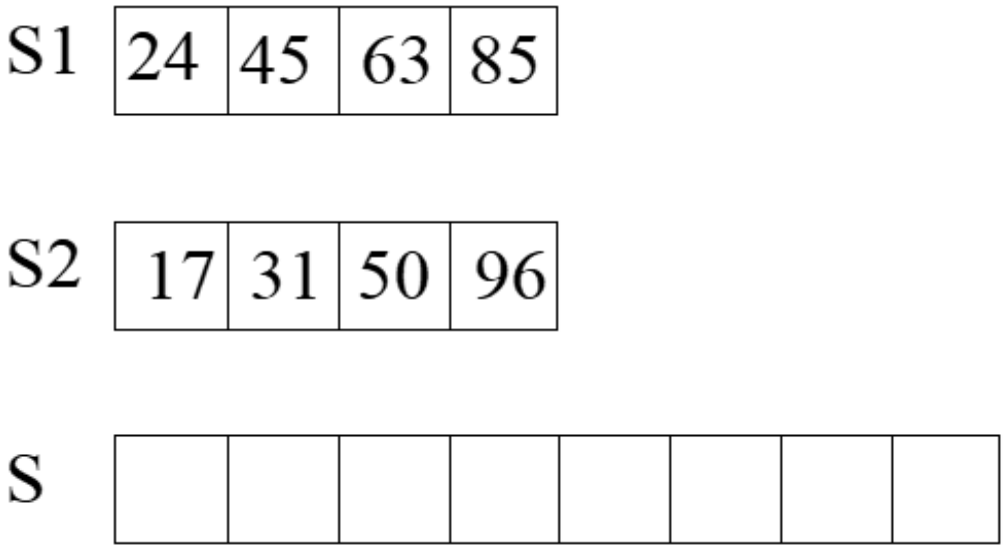


Mezcla (Merge)



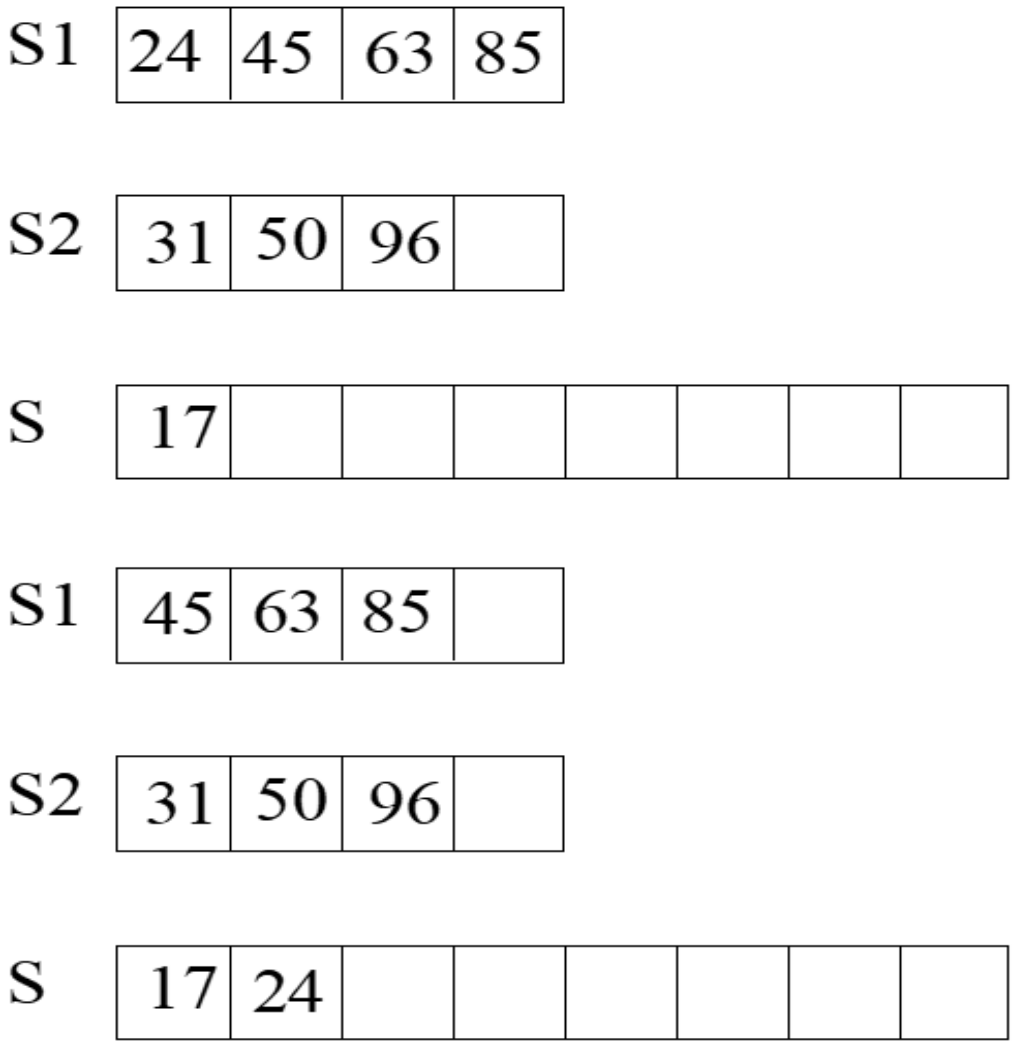
Ejemplo de mezcla para dos secuencias ordenadas

Mezcla (Merge)



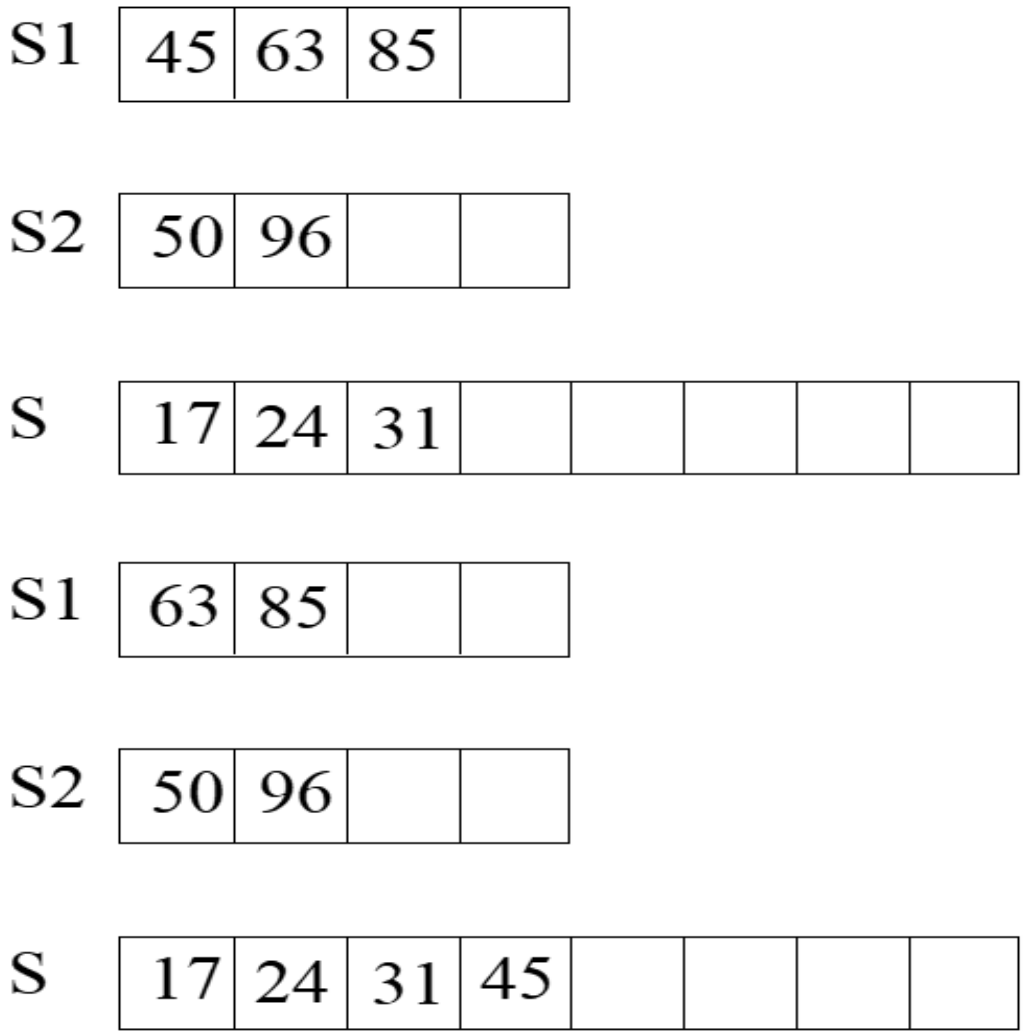
Ejemplo de mezcla para dos secuencias ordenadas

Mezcla (Merge)



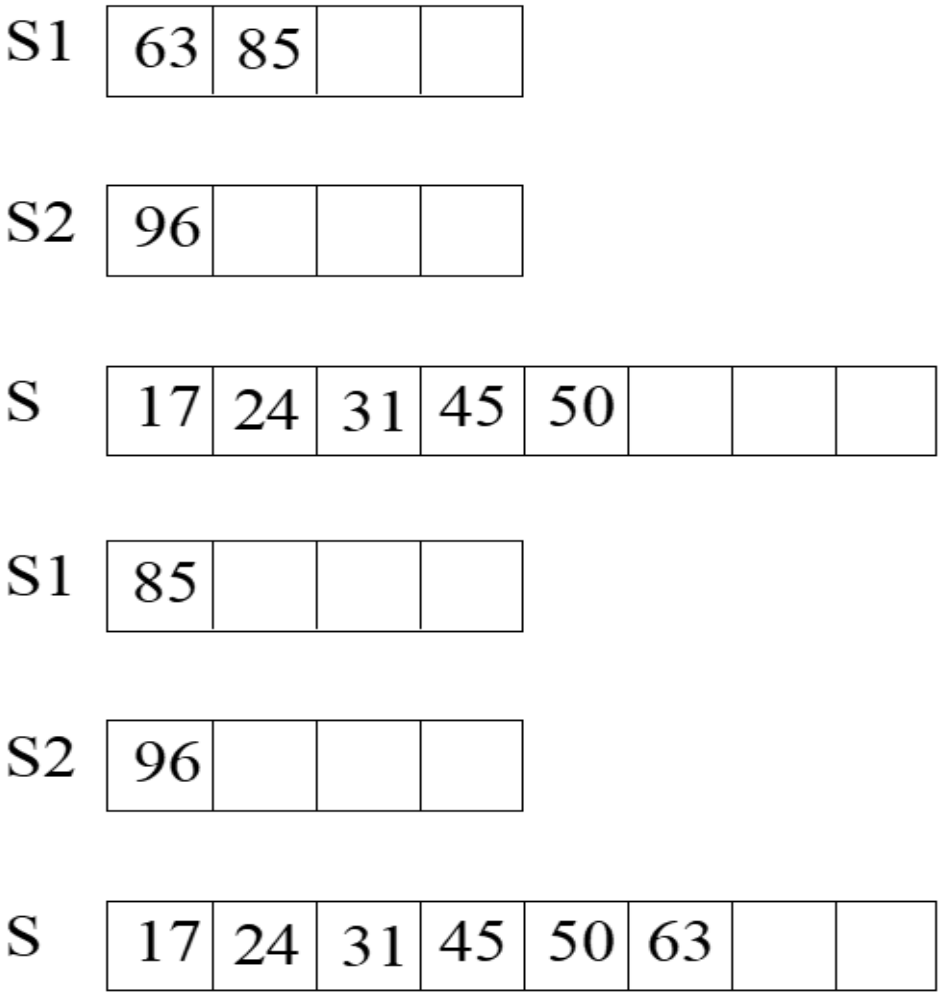
Ejemplo de mezcla para dos secuencias ordenadas

Mezcla (Merge)



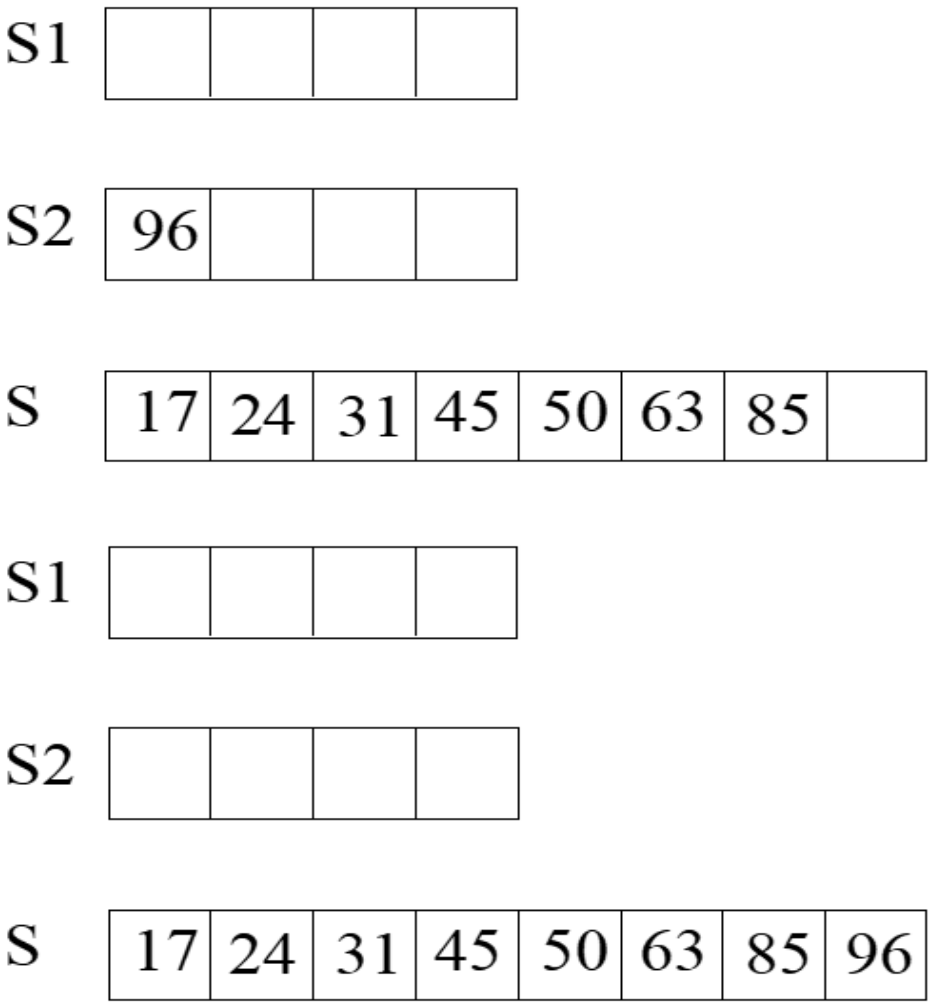
Ejemplo de mezcla para dos secuencias ordenadas

Mezcla (Merge)



Ejemplo de mezcla para dos secuencias ordenadas

Mezcla (Merge)



Complejidad del Algoritmo MergeSort

Tiempo de ejecución de Merge Sort

El tiempo de ejecución τ de la mezcla (Merge) es proporcional a $nlog(n)$ donde n es el tamaño de la secuencia.

$$\tau = O(nlog(n))$$

Recuerde que el tiempo de funcionamiento del algoritmo de burbuja es proporcional al cuadrado de n $O(n^2)$. Por lo tanto, MergeSort es mucho más rápido que BubbleSort.


Insertion
Selection
Bubble
Shell
Merge
Heap
Quick
Quick3
Random
Nearly Sorted
Reversed
Few Unique

Referencias I

Thomas H Cormen.

Introduction to algorithms.

MIT press, 2009.

Donald Knuth.

The art of programming.

ITNow, 53(4), 2011.

Elba Karen Sáenz García.

Guía práctica de estudio 1, algoritmos de ordenamiento parte 1.

Facultad de ingeniería, UNAM.