

Algoritmos de Ordenamiento

Marco Antonio Heredia



Posgrado en
Optimización

Universidad
Autónoma
Metropolitana

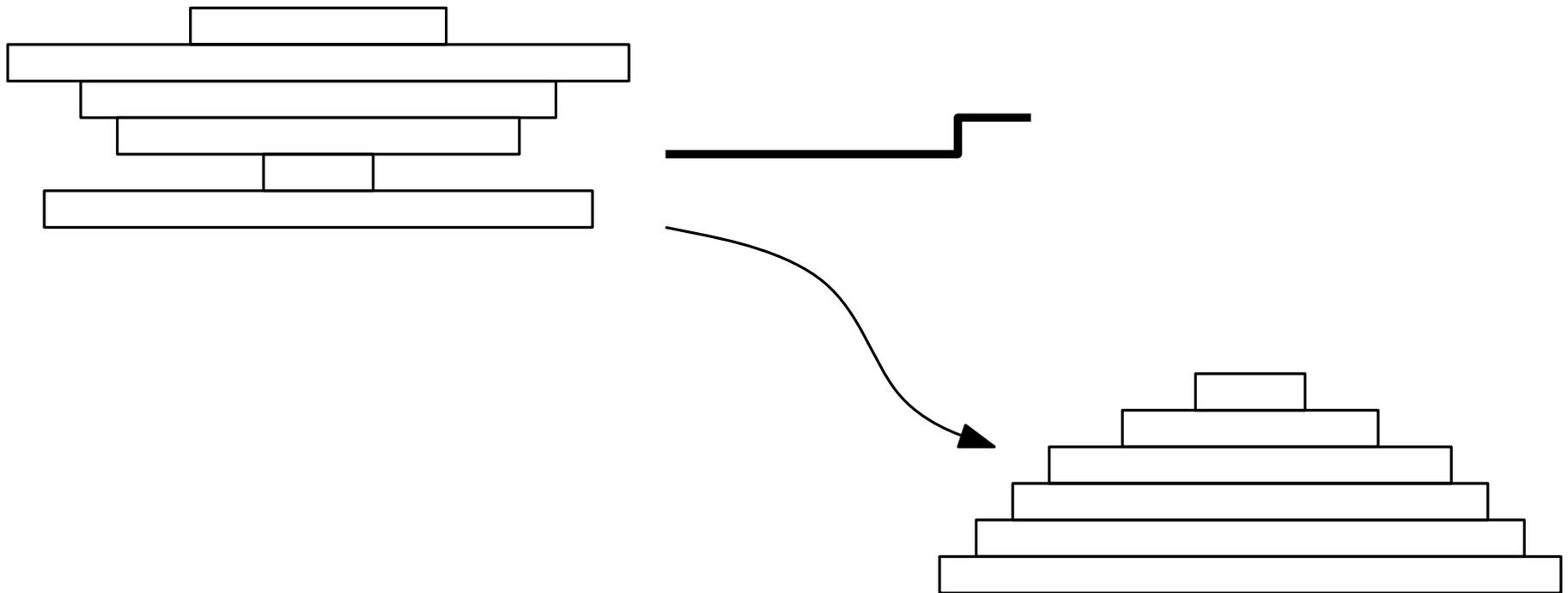


Casa abierta al tiempo **Azcapotzalco**

Ordenar

“Acomodar” los elementos de un conjunto de acuerdo a cierto orden (numérico, alfabético, tamaño, etc.).

- ▶ Por lo general dicho orden es total y los elementos se acomodan de “menor” a “mayor”.



Ordenamiento interno

- ▶ Se tiene acceso a cualquiera de los elementos a ordenar durante todo el proceso de ordenamiento.
- ▶ En el ordenamiento externo, sólo se tiene acceso a partes del conjunto a ordenar, en cada momento. Por ejemplo, cuando la cantidad de números no caben juntos dentro de la memoria RAM.

Ideas previas

- ▶ Por facilidad de exposición, ordenaremos conjuntos de n números enteros, de menor a mayor.
- ▶ Asumiremos ordenamiento interno, aunque algunos algoritmos también funcionan para el caso externo.
- ▶ La serie aritmética o *suma de Gauss* tendrá un papel importante:

$$1 + 2 + \dots + k = \sum_{i=1}^k i = \frac{k(k+1)}{2}$$

Selection Sort (Ordenamiento por Selección)

Selection sort

15	8	5	2	-1	3	9
----	---	---	---	----	---	---

- ▶ Buscar el mínimo elemento de tu arreglo actual.

Selection sort

15	8	5	2	-1	3	9
----	---	---	---	----	---	---



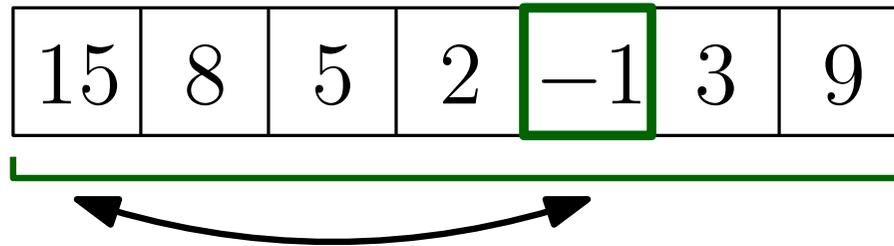
- ▶ Buscar el mínimo elemento de tu arreglo actual.

Selection sort

15	8	5	2	-1	3	9
----	---	---	---	----	---	---

- ▶ Buscar el mínimo elemento de tu arreglo actual.

Selection sort



- ▶ Buscar el mínimo elemento de tu arreglo actual.
- ▶ Poner ese mínimo al principio del arreglo (intercambiar).

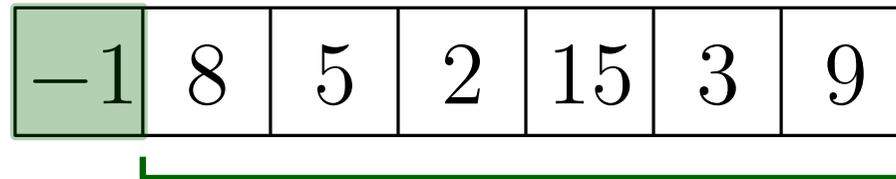
Selection sort

-1	8	5	2	15	3	9
----	---	---	---	----	---	---

- ▶ Buscar el mínimo elemento de tu arreglo actual.
- ▶ Poner ese mínimo al principio del arreglo (intercambiar).

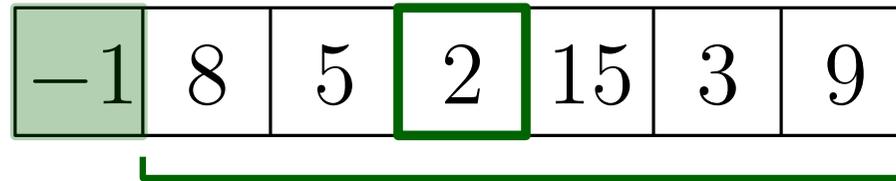
Selection sort

-1	8	5	2	15	3	9
----	---	---	---	----	---	---



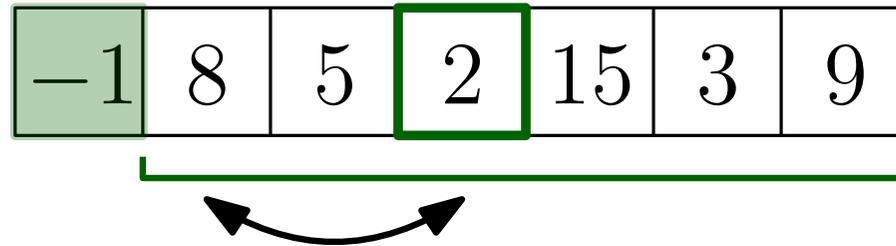
- ▶ Buscar el mínimo elemento de tu arreglo actual.
- ▶ Poner ese mínimo al principio del arreglo (intercambiar).
- ▶ Considerar el arreglo sin la primera casilla y repetir hasta que no falten números por ordenar.

Selection sort



- ▶ Buscar el mínimo elemento de tu arreglo actual.
- ▶ Poner ese mínimo al principio del arreglo (intercambiar).
- ▶ Considerar el arreglo sin la primera casilla y repetir hasta que no falten números por ordenar.

Selection sort



- ▶ Buscar el mínimo elemento de tu arreglo actual.
- ▶ Poner ese mínimo al principio del arreglo (intercambiar).
- ▶ Considerar el arreglo sin la primera casilla y repetir hasta que no falten números por ordenar.

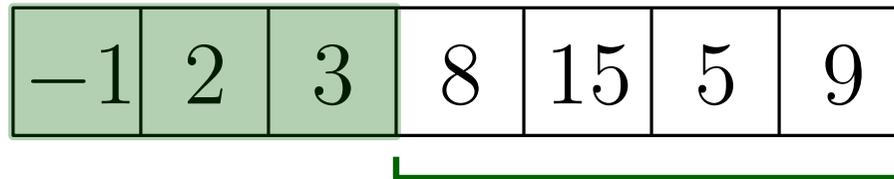
Selection sort

-1	2	5	8	15	3	9
----	---	---	---	----	---	---

- ▶ Buscar el mínimo elemento de tu arreglo actual.
- ▶ Poner ese mínimo al principio del arreglo (intercambiar).
- ▶ Considerar el arreglo sin la primera casilla y repetir hasta que no falten números por ordenar.

Selection sort

-1	2	3	8	15	5	9
----	---	---	---	----	---	---



- ▶ Buscar el mínimo elemento de tu arreglo actual.
- ▶ Poner ese mínimo al principio del arreglo (intercambiar).
- ▶ Considerar el arreglo sin la primera casilla y repetir hasta que no falten números por ordenar.

Selection sort

-1	2	3	5	15	8	9
----	---	---	---	----	---	---



- ▶ Buscar el mínimo elemento de tu arreglo actual.
- ▶ Poner ese mínimo al principio del arreglo (intercambiar).
- ▶ Considerar el arreglo sin la primera casilla y repetir hasta que no falten números por ordenar.

Selection sort

-1	2	3	5	8	9	15
----	---	---	---	---	---	----

- ▶ Buscar el mínimo elemento de tu arreglo actual.
- ▶ Poner ese mínimo al principio del arreglo (intercambiar).
- ▶ Considerar el arreglo sin la primera casilla y repetir hasta que no falten números por ordenar.

Selection sort

Tamaño
arreglo
actual

Encontrar el mínimo e intercambiar

n

n operaciones

$n - 1$

$n - 1$ operaciones

$n - 2$

$n - 2$ operaciones

\vdots

\vdots

2

2 operaciones

1

1 operación

Selection sort

Tamaño
arreglo
actual

Encontrar el mínimo e intercambiar

n

n operaciones

$n - 1$

$n - 1$ operaciones

$n - 2$

$n - 2$ operaciones

\vdots

\vdots

2

2 operaciones

1

1 operación

$$n + (n - 1) + \dots + 2 + 1 = \sum_{i=1}^n i = \frac{n(n + 1)}{2} = O(n^2)$$

Selection sort

```
int idx_min=0, i, j;
int swap;
for(i=0 ; i<n-1 ; i++)
{
    idx_min=i;
    for(j=i+1 ; j<n ; j++) {
        if (A[idx_min] > A[j])
            idx_min=j; }
    swap=A[idx_min];
    A[idx_min]=A[i];
    A[i]=swap;
}
```

Bubble Sort (Ordenamiento Burbuja)

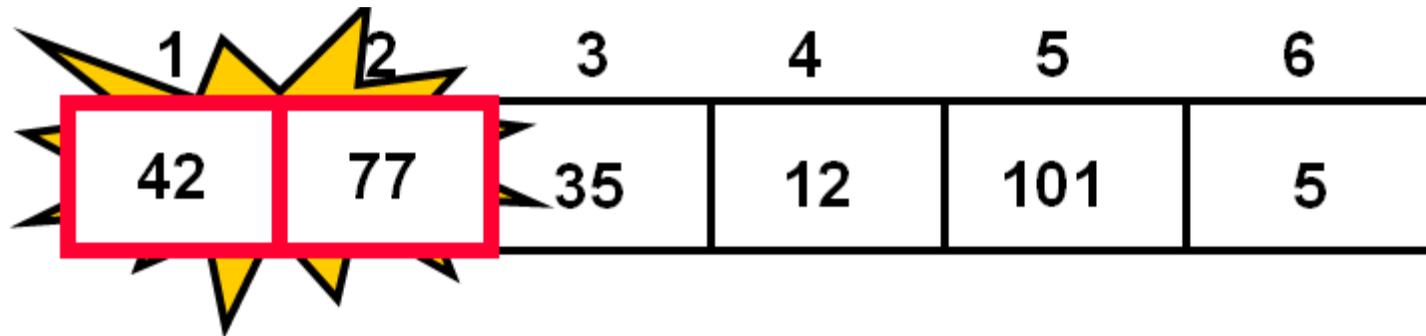
Bubble sort

- ▶ En el arreglo actual, comparar por parejas contiguas de forma ascendente.
- ▶ Con cada comparación los números más grandes “flotan” hacia “arriba” .
- ▶ Al final de la etapa, el número máximo llegó al final del arreglo.
- ▶ Considerar el arreglo sin la última casilla y repetir hasta que no falten números por ordenar

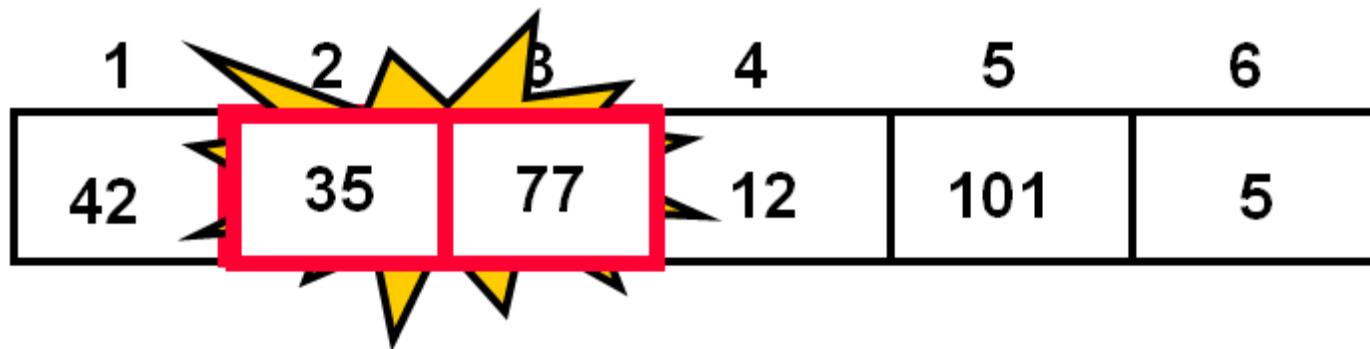
Bubble sort

1	2	3	4	5	6
77	42	35	12	101	5

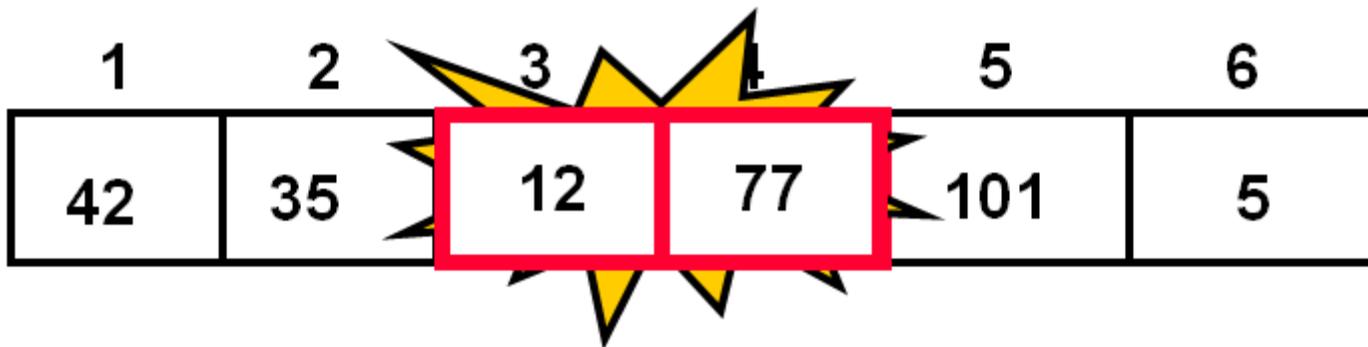
Bubble sort



Bubble sort



Bubble sort

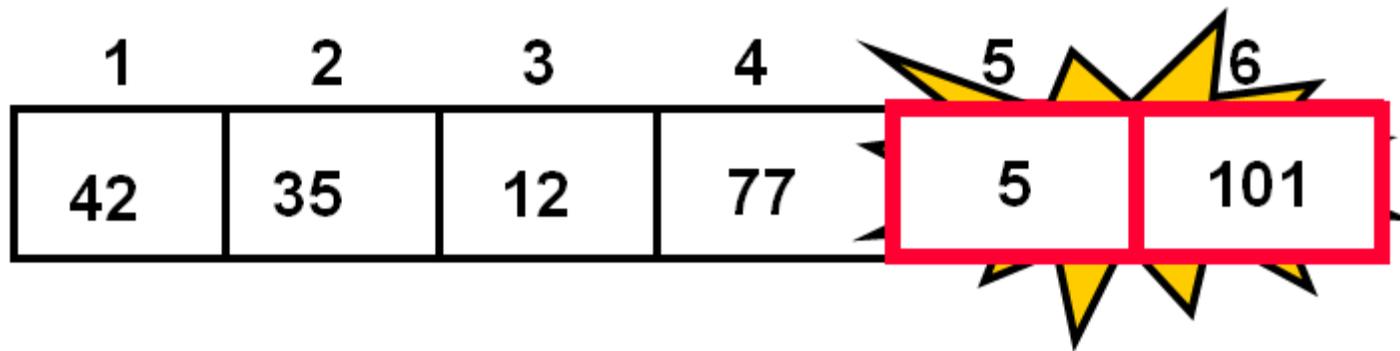


Bubble sort

1	2	3	4	5	6
42	35	12	77	101	5

No necesita intercambio

Bubble sort



Bubble sort

1	2	3	4	5	6
42	35	12	77	5	101

El máximo alcanzó su lugar

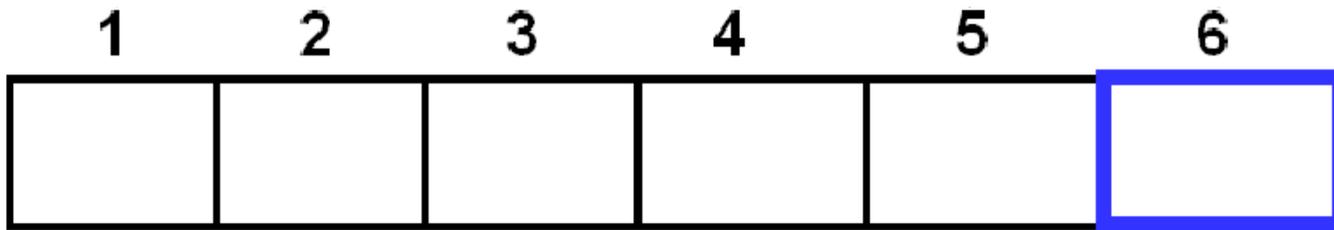
Bubble sort

1	2	3	4	5	6
42	35	12	77	5	101

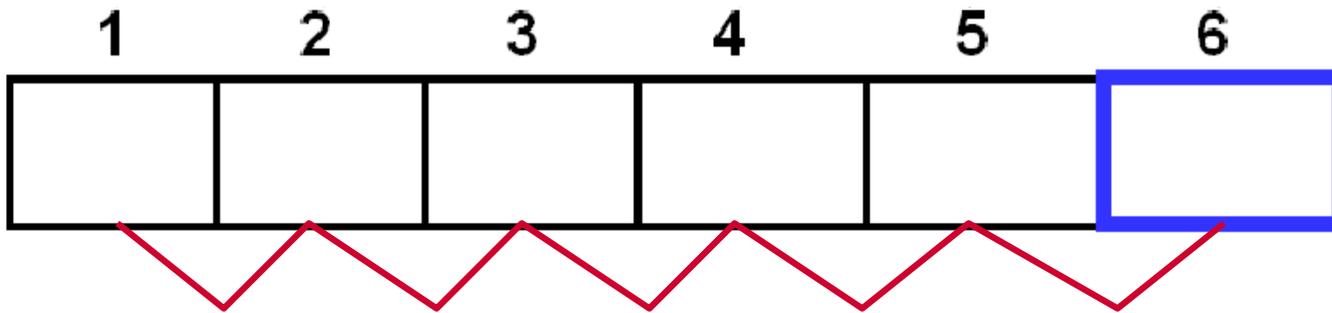
El máximo alcanzó su lugar

Aquí es donde se repite el proceso en el arreglo restante.

Bubble sort



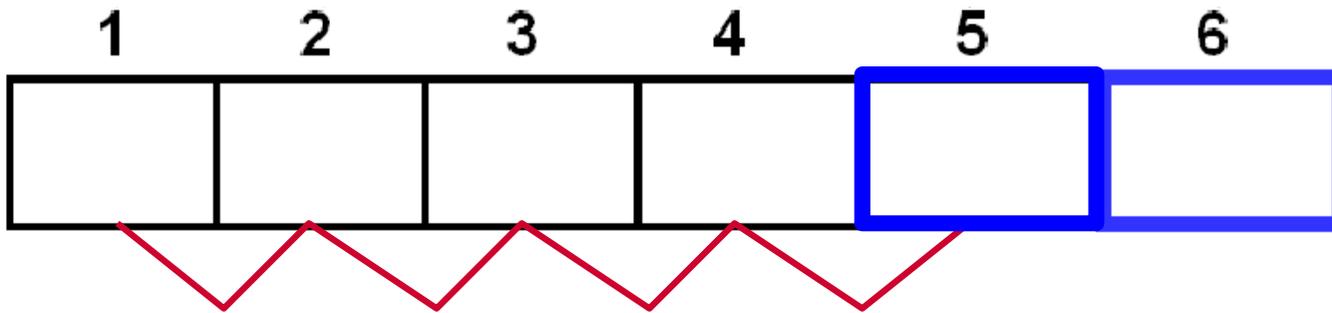
Bubble sort



► 1a etapa

operaciones (peor caso)
 $n-1$ intercambios

Bubble sort



operaciones (peor caso)

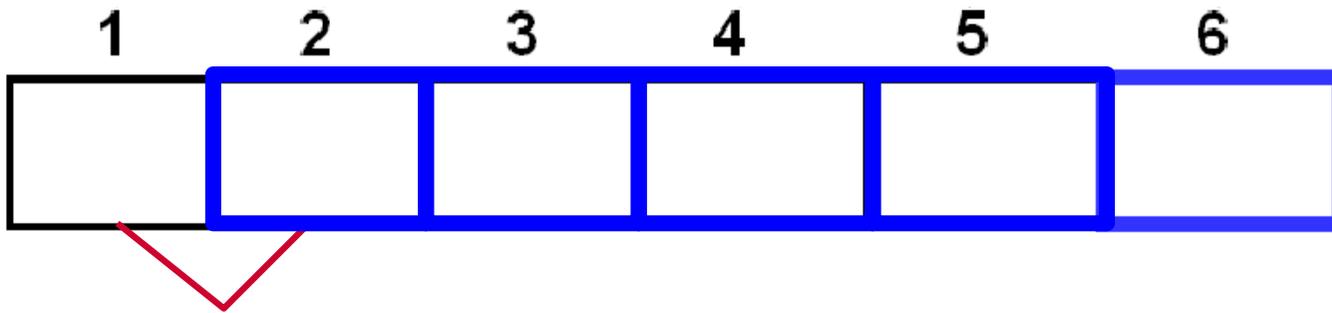
► 1a etapa

$n-1$ intercambios

► 2a etapa

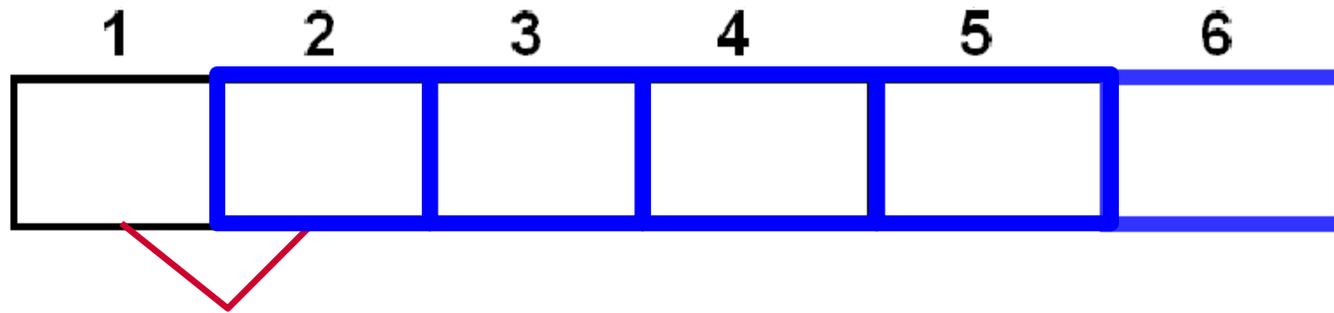
$n-2$ intercambios

Bubble sort



- | | |
|-------------------|-------------------------|
| | operaciones (peor caso) |
| ▶ 1a etapa | $n-1$ intercambios |
| ▶ 2a etapa | $n-2$ intercambios |
| ⋮ | ⋮ |
| ▶ $(n-1)$ a etapa | 1 intercambio |

Bubble sort



operaciones (peor caso)

- ▶ 1a etapa $n-1$ intercambios
- ▶ 2a etapa $n-2$ intercambios
- ⋮ ⋮
- ▶ $(n-1)$ a etapa 1 intercambio

$$(n-1) + (n-2) + \dots + 1 = \sum_{i=1}^{n-1} i = \frac{(n-1)(n)}{2} = O(n^2)$$

Bubble sort

```
int swap, i, j;

for (i = n-1; i >0 ; i--) {
    for (j = 0; j < i ; j++) {
        if (A[j] > A[j+1]) {
            swap = A[j+1];
            A[j+1] = A[j];
            A[j] = swap;
        }
    }
}
```

Insertion Sort (Ordenamiento por Inserción)

Insertion sort

15	8	5	9	6	-1	2
----	---	---	---	---	----	---

1. Asumir que la casilla inicial ya está ordenada.

Insertion sort

15	8	5	9	6	-1	2
----	---	---	---	---	----	---

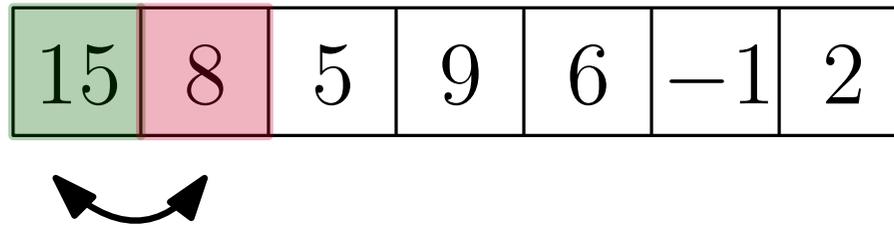
1. Asumir que la casilla inicial ya está ordenada.

Insertion sort

15	8	5	9	6	-1	2
----	---	---	---	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.

Insertion sort



1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.

Insertion sort

8	15	5	9	6	-1	2
---	----	---	---	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.

Insertion sort

8	15	5	9	6	-1	2
---	----	---	---	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

8	15	5	9	6	-1	2
---	----	---	---	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

8	5	15	9	6	-1	2
---	---	----	---	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

5	8	15	9	6	-1	2
---	---	----	---	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

5	8	15	9	6	-1	2
---	---	----	---	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

5	8	15	9	6	-1	2
---	---	----	---	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

5	8	9	15	6	-1	2
---	---	---	----	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

5	8	9	15	6	-1	2
---	---	---	----	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

5	6	8	9	15	-1	2
---	---	---	---	----	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

5	6	8	9	15	-1	2
---	---	---	---	----	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

-1	5	6	8	9	15	2
----	---	---	---	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

-1	5	6	8	9	15	2
----	---	---	---	---	----	---

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

-1	2	5	6	8	9	15
----	---	---	---	---	---	----

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

-1	2	5	6	8	9	15
----	---	---	---	---	---	----

1. Asumir que la casilla inicial ya está ordenada.
2. Tomar el primer elemento de la región no-ordenada.
3. Intercambiar ese elemento, hacia la izquierda, con los elementos de la región ordenada, hasta que se *inserte* en la posición adecuada.
4. Repetir desde 2. hasta que no falten números por ordenar.

Insertion sort

	operaciones (peor caso)
▶ Paso 1.	0 intercambios
▶ 1a etapa	1 intercambio
▶ 2a etapa	2 intercambios
▶ ⋮	⋮
▶ $(n-1)$ a etapa	$n-1$ intercambios

Insertion sort

	operaciones (peor caso)
▶ Paso 1.	0 intercambios
▶ 1a etapa	1 intercambio
▶ 2a etapa	2 intercambios
▶ ⋮	⋮
▶ $(n-1)$ a etapa	$n-1$ intercambios

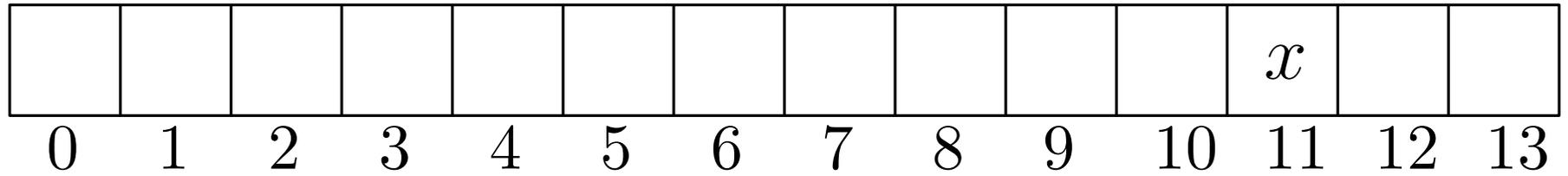
$$0 + 1 + 2 \dots + (n-1) = \sum_{i=1}^{n-1} i = \frac{(n-1)(n)}{2} = O(n^2)$$

Quick Sort (Ordenamiento Rápido)

Quick sort

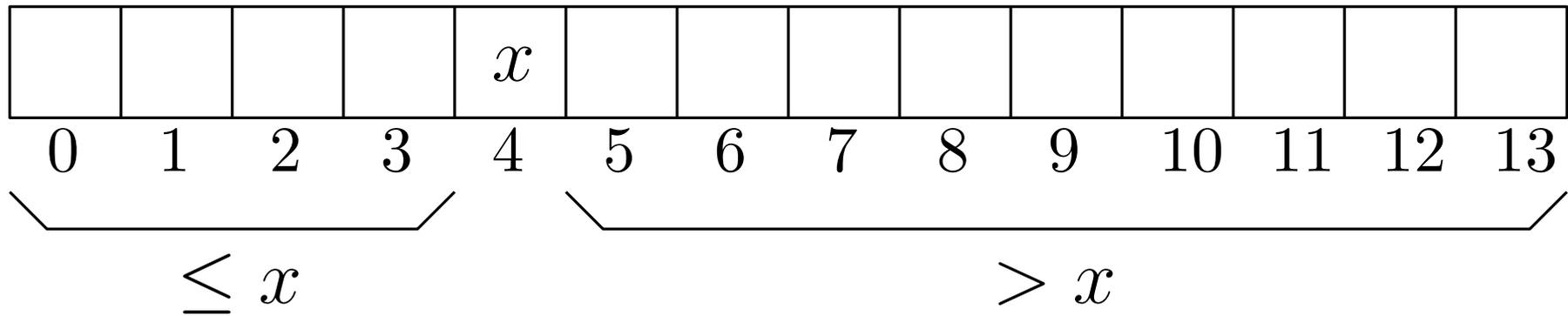
0	1	2	3	4	5	6	7	8	9	10	11	12	13	

Quick sort



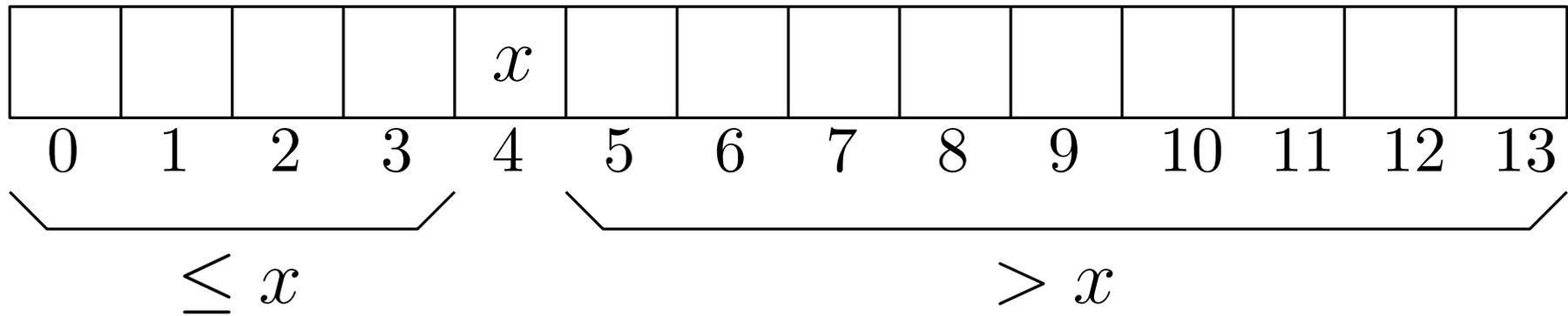
- ▶ Escoge aleatoriamente un elemento Pivote, x .

Quick sort



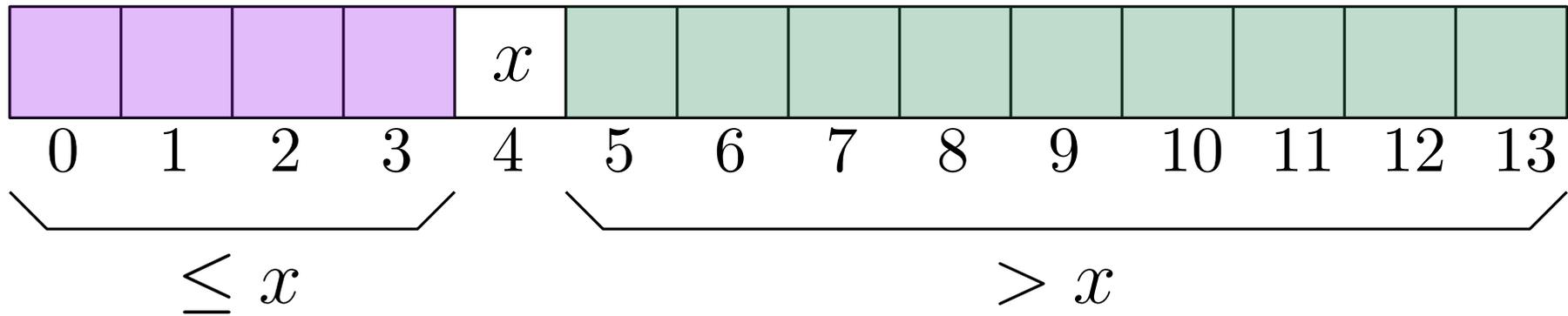
- ▶ Escoge aleatoriamente un elemento Pivote, x .
- ▶ Acomoda los elementos del arreglo, para que los elementos menores que x queden a su izquierda (sin ordenar), y los mayores queden a su derecha (sin ordenar).

Quick sort



- ▶ Escoge aleatoriamente un elemento Pivote, x .
- ▶ Acomoda los elementos del arreglo, para que los elementos menores que x queden a su izquierda (sin ordenar), y los mayores queden a su derecha (sin ordenar).
- ▶ Nota que el Pivote ya llegó a su posición final.

Quick sort



- ▶ Escoge aleatoriamente un elemento Pivote, x .
- ▶ Acomoda los elementos del arreglo, para que los elementos menores que x queden a su izquierda (sin ordenar), y los mayores queden a su derecha (sin ordenar).
- ▶ Nota que el Pivote ya llegó a su posición final.
- ▶ Ordena los elementos del lado izquierdo con Quick Sort y los del lado derecho con Quick Sort (2 llamadas recursivas).

Quick sort

15	7	-1	5	2	8	-1	8	3	-6	6	9	31	17
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Complejidad de Quick Sort

Quick Sort  $O(n^2)$ en el peor caso
 $O(n \log n)$ en promedio

Complejidad de Quick Sort

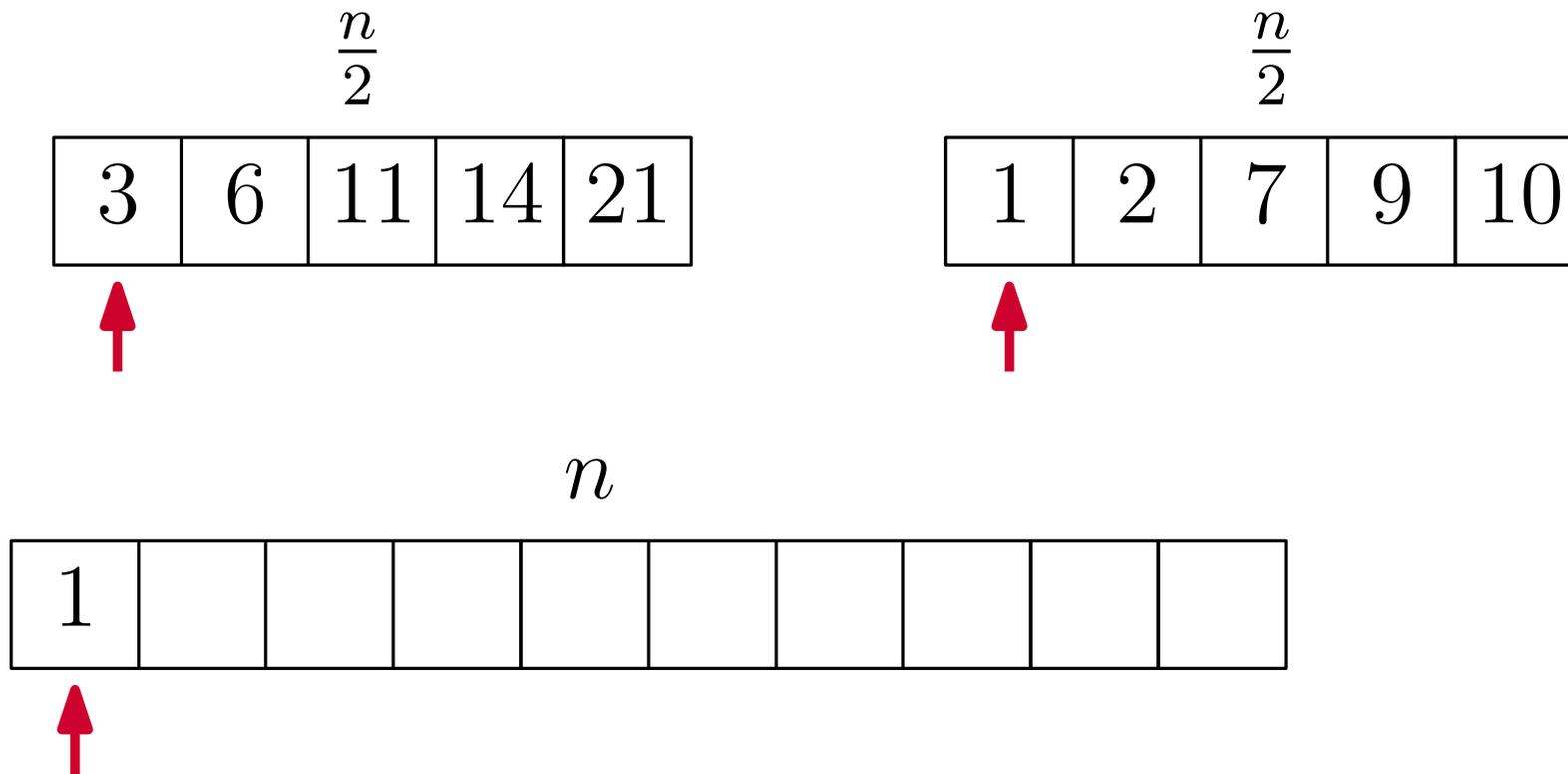
Quick Sort $\left\{ \begin{array}{l} O(n^2) \text{ en el peor caso} \\ O(n \log n) \text{ en promedio} \end{array} \right.$

Nota: Que el peor caso ocurra es muy poco probable

Merge Sort (Ordenamiento por Mezcla)

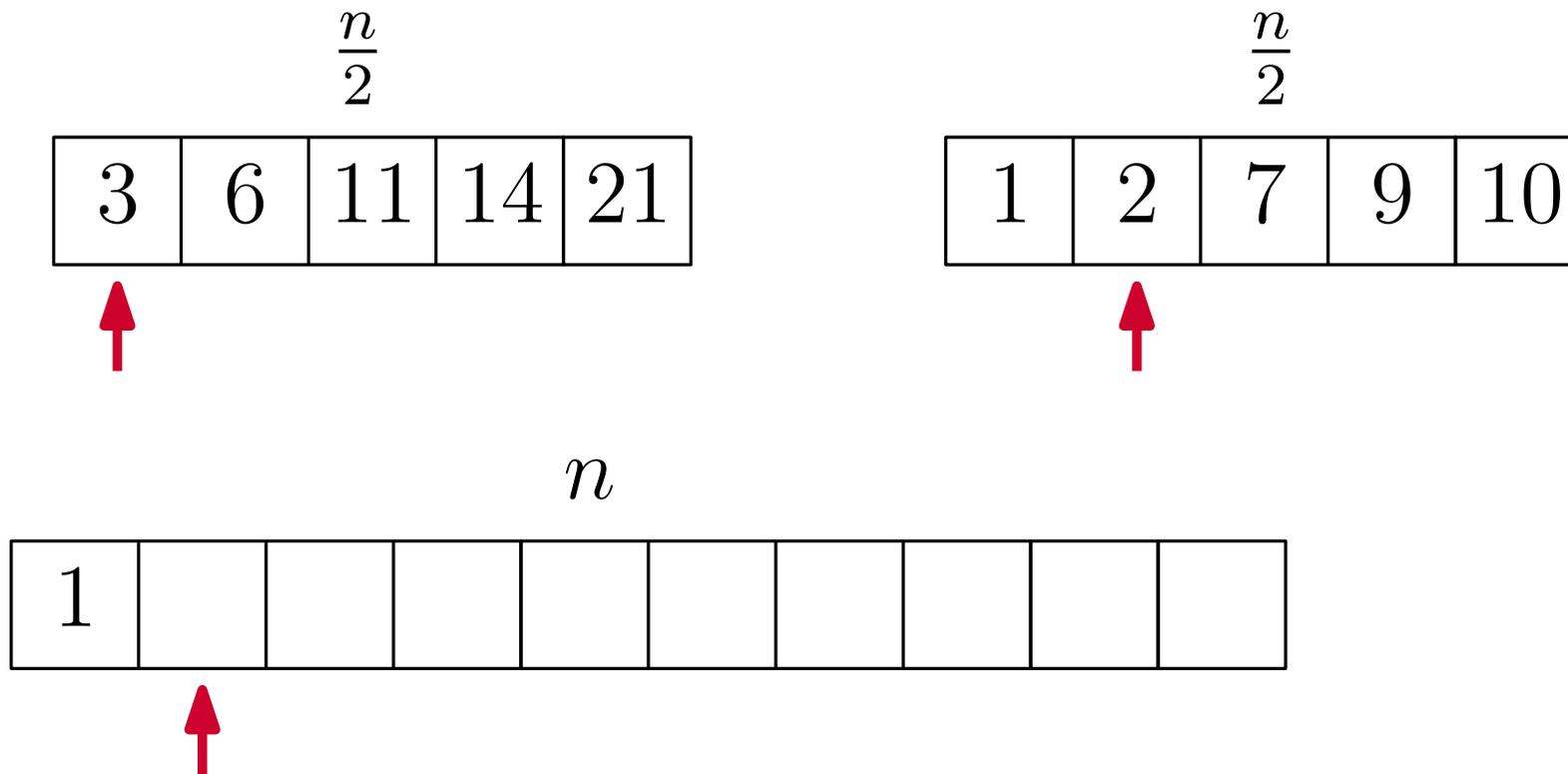
¿A qué le llamamos Merge (Mezcla)?

Dados 2 arreglos ordenados, mezclarlos en 1 sólo arreglo ordenado.



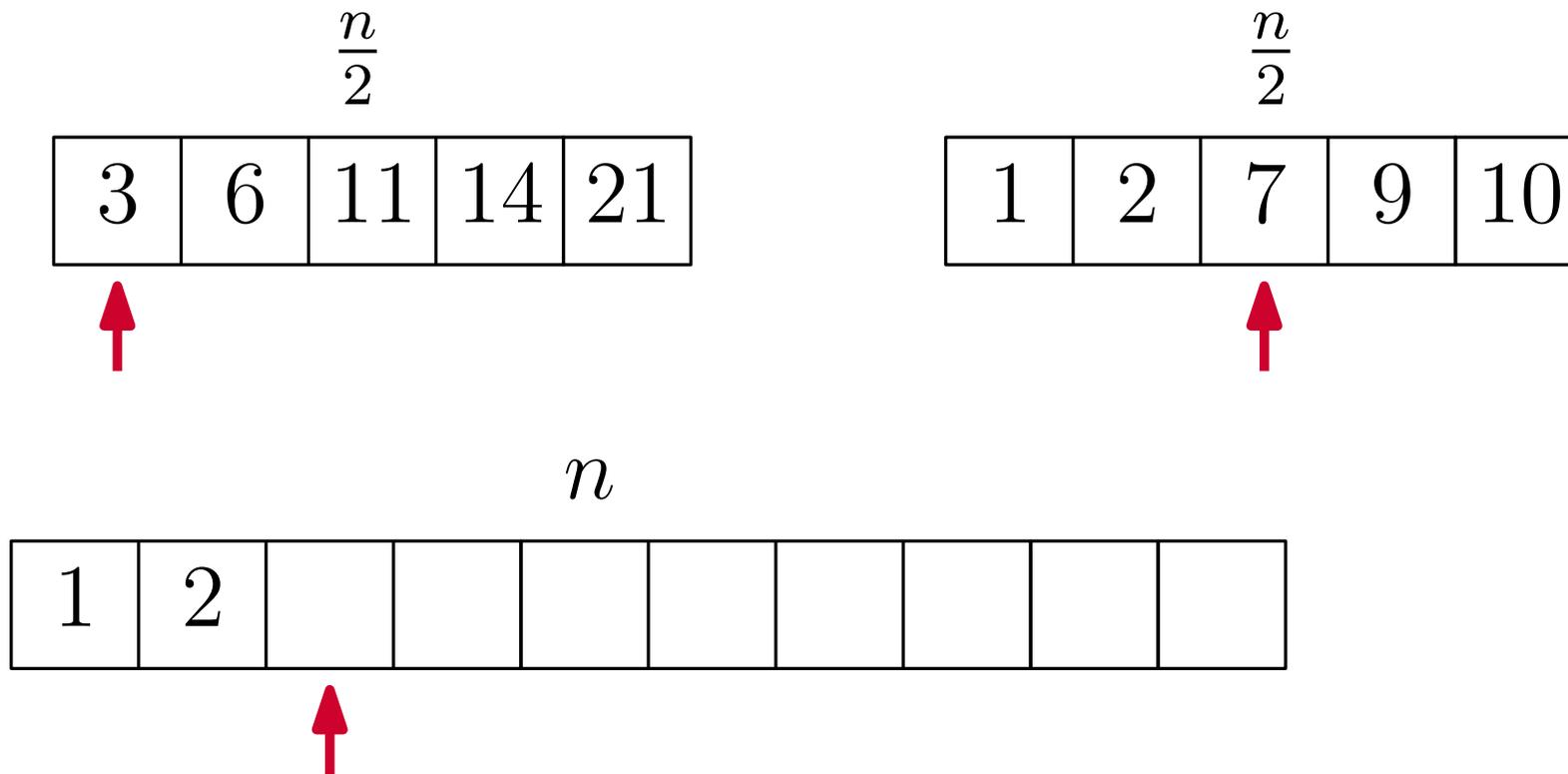
¿A qué le llamamos Merge (Mezcla)?

Dados 2 arreglos ordenados, mezclarlos en 1 sólo arreglo ordenado.



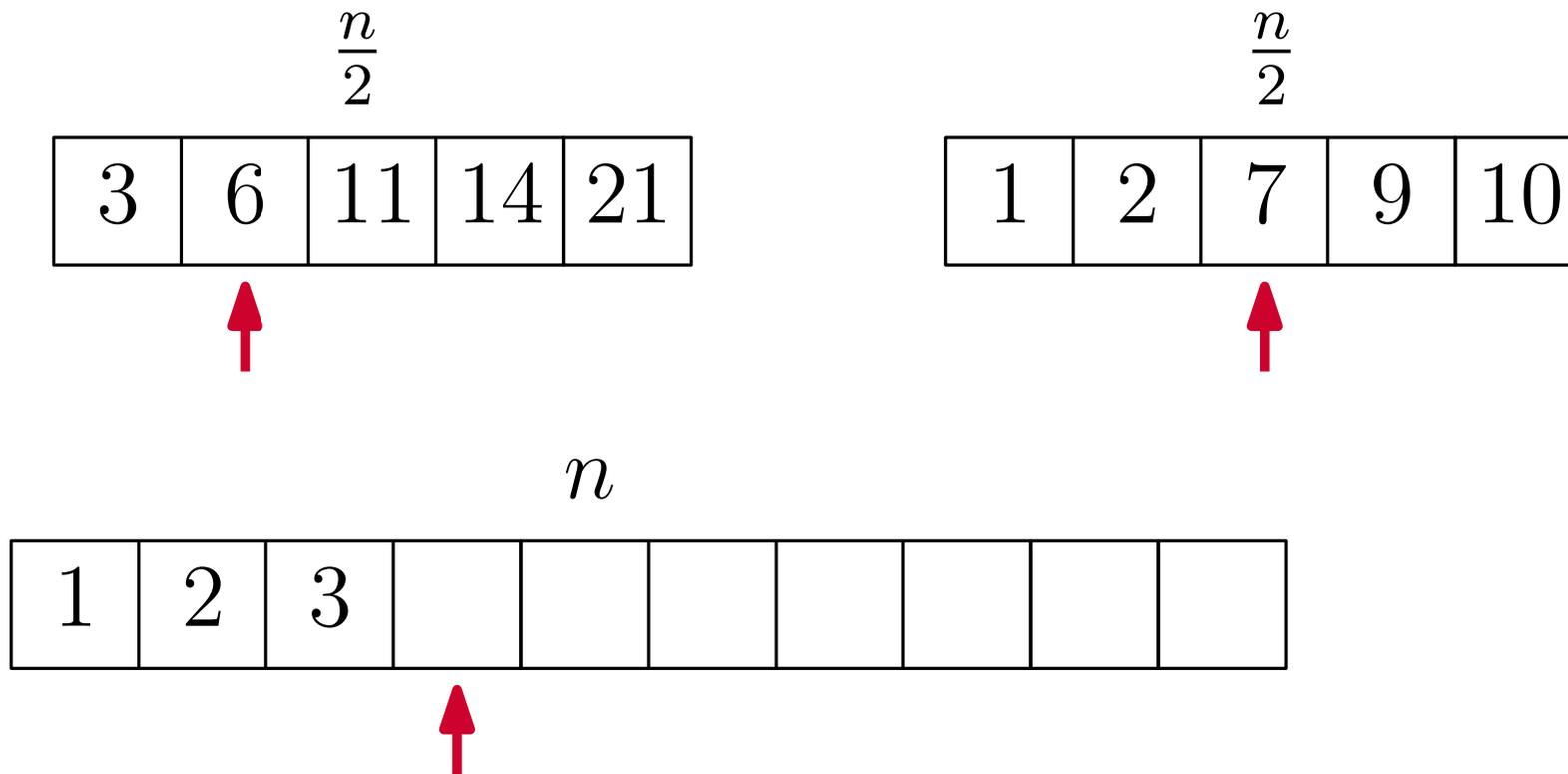
¿A qué le llamamos Merge (Mezcla)?

Dados 2 arreglos ordenados, mezclarlos en 1 sólo arreglo ordenado.



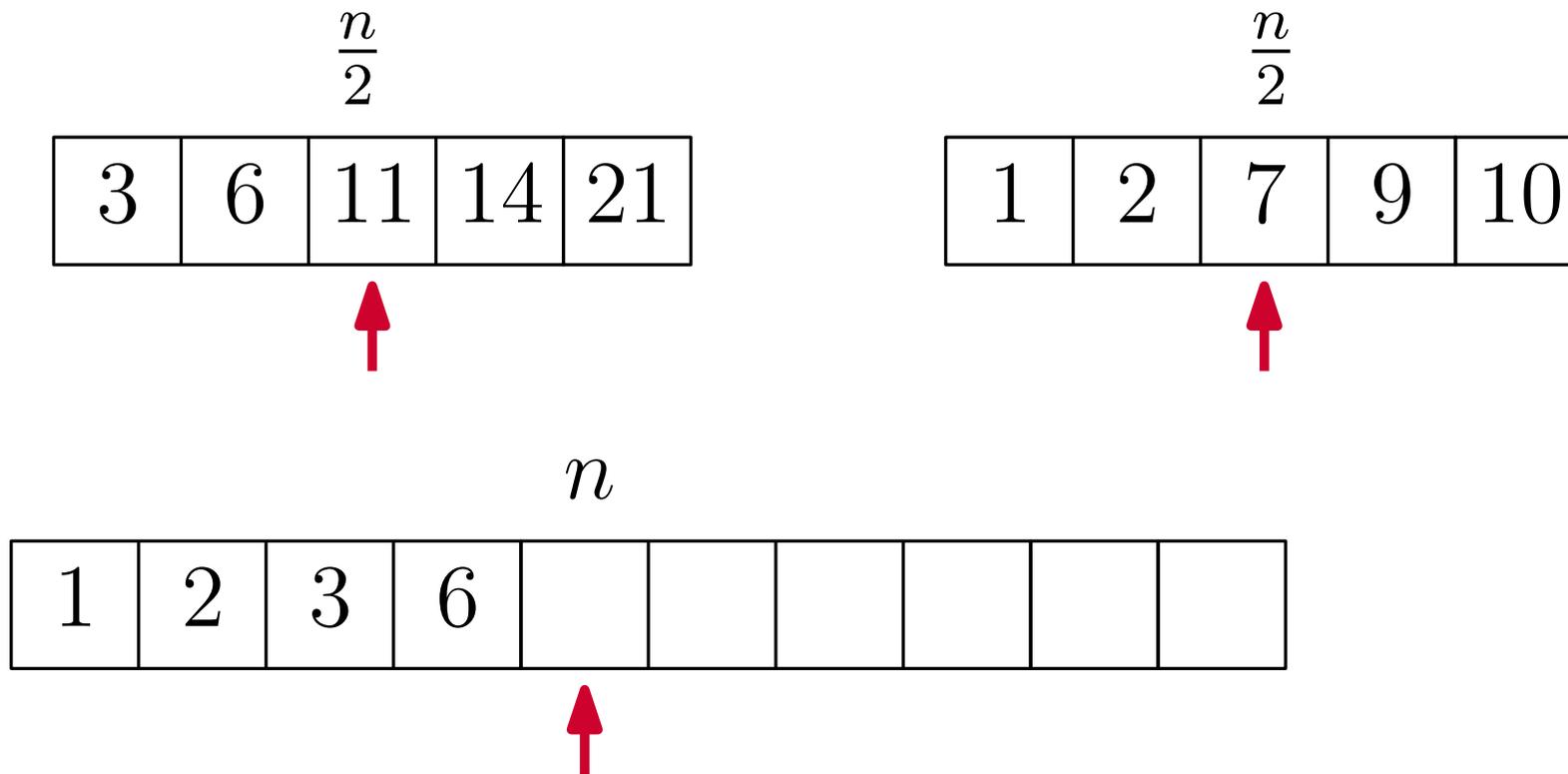
¿A qué le llamamos Merge (Mezcla)?

Dados 2 arreglos ordenados, mezclarlos en 1 sólo arreglo ordenado.



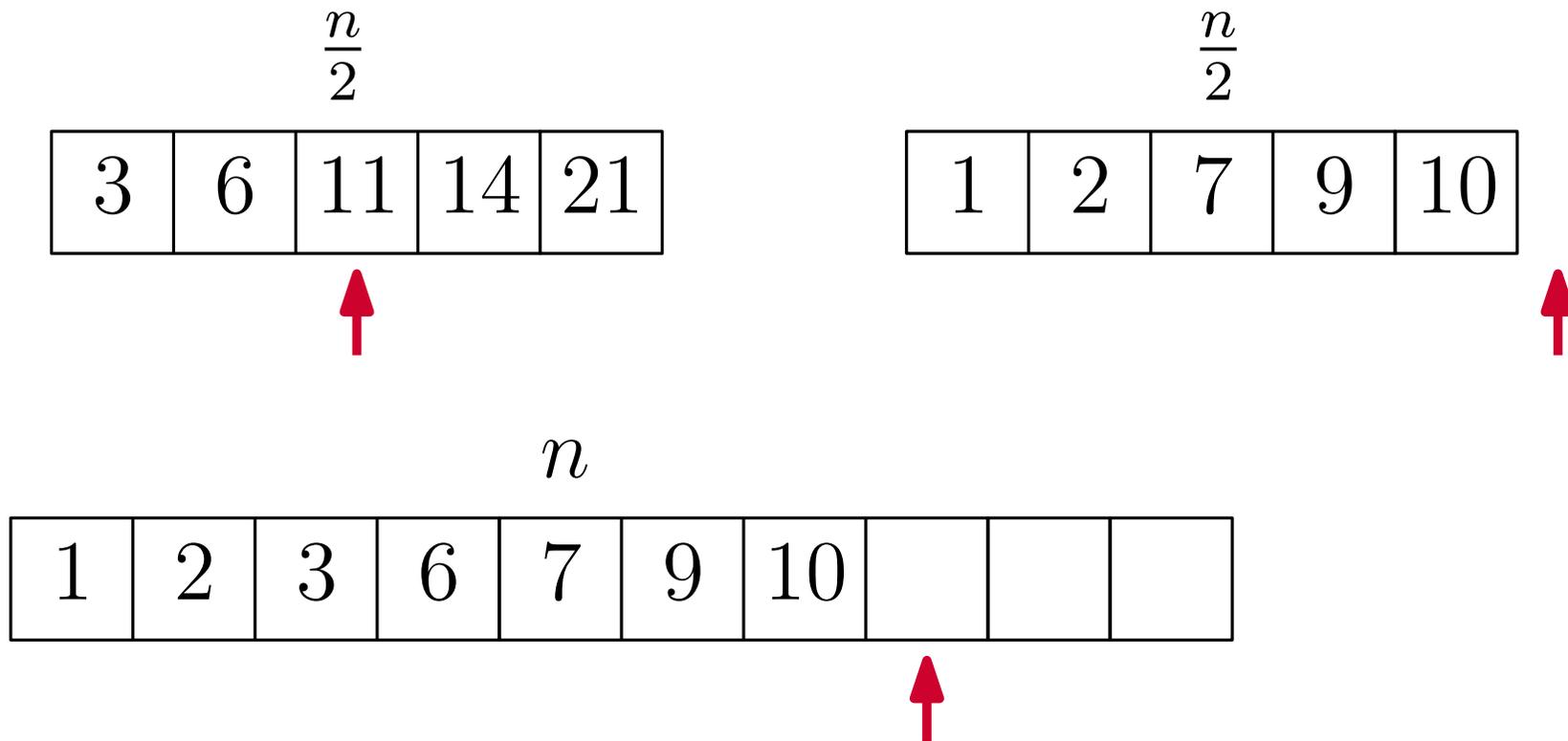
¿A qué le llamamos Merge (Mezcla)?

Dados 2 arreglos ordenados, mezclarlos en 1 sólo arreglo ordenado.



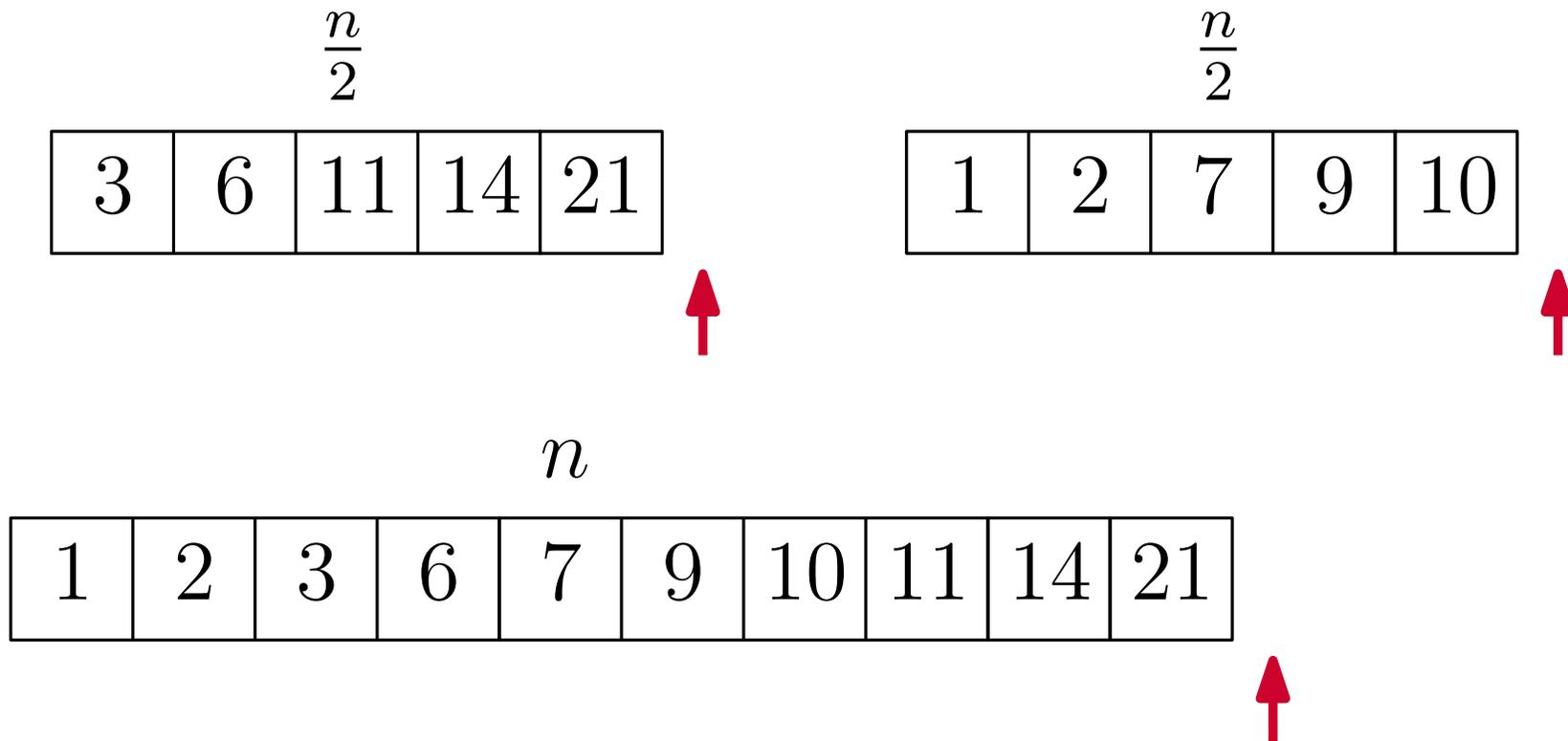
¿A qué le llamamos Merge (Mezcla)?

Dados 2 arreglos ordenados, mezclarlos en 1 sólo arreglo ordenado.



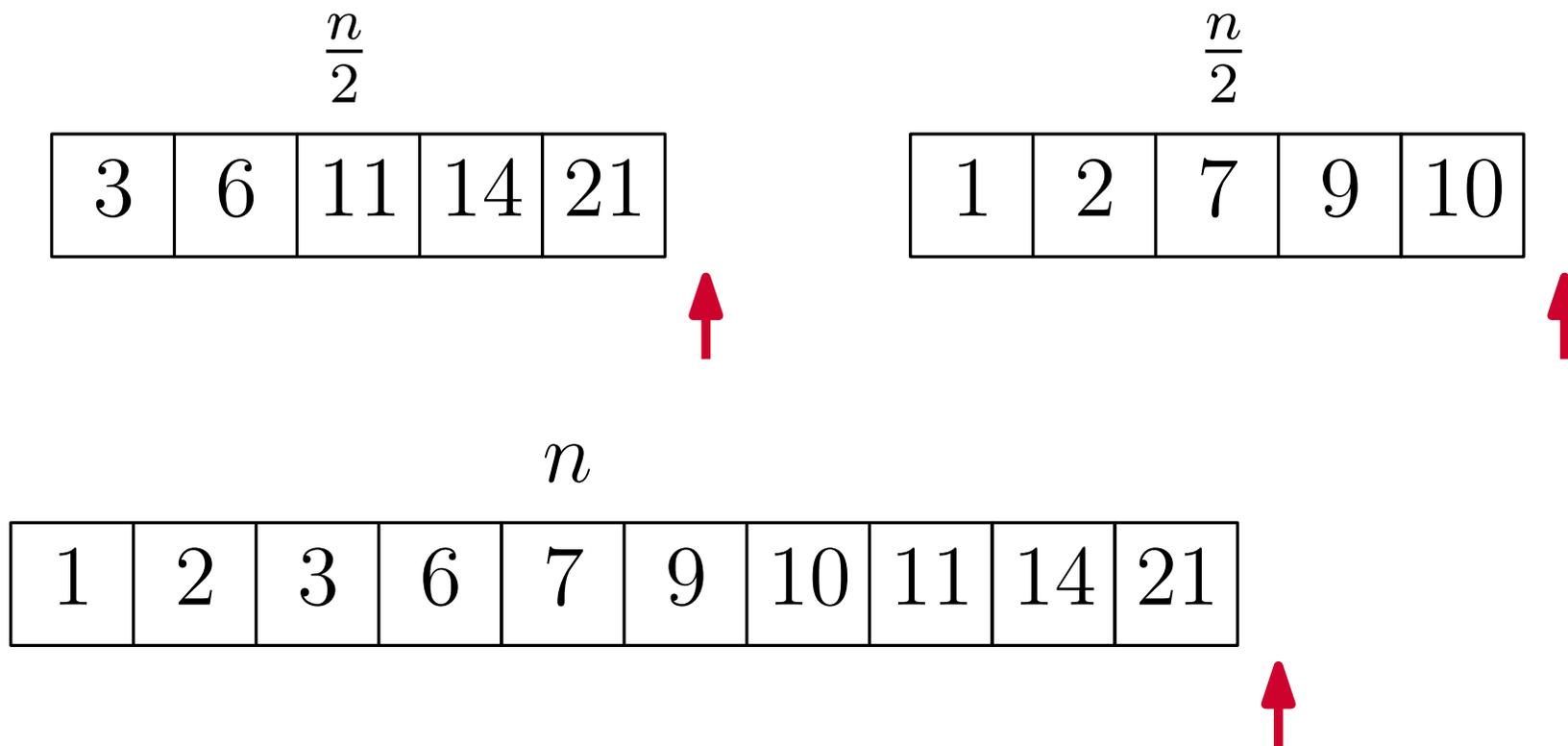
¿A qué le llamamos Merge (Mezcla)?

Dados 2 arreglos ordenados, mezclarlos en 1 sólo arreglo ordenado.



¿A qué le llamamos Merge (Mezcla)?

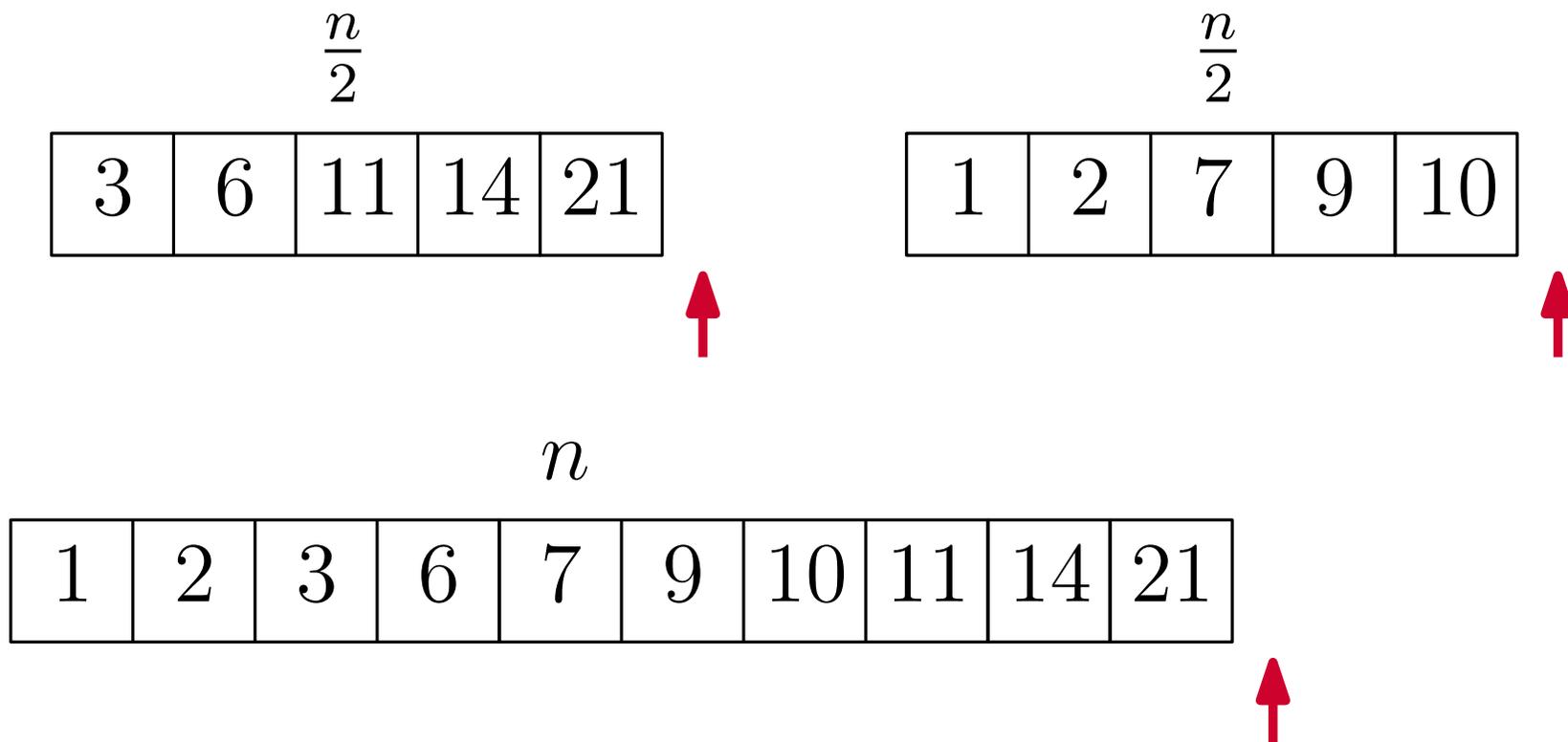
Dados 2 arreglos ordenados, mezclarlos en 1 sólo arreglo ordenado.



En cada paso, hubo un movimiento de flecha a la derecha en alguno de los arreglos originales.

¿A qué le llamamos Merge (Mezcla)?

Dados 2 arreglos ordenados, mezclarlos en 1 sólo arreglo ordenado.



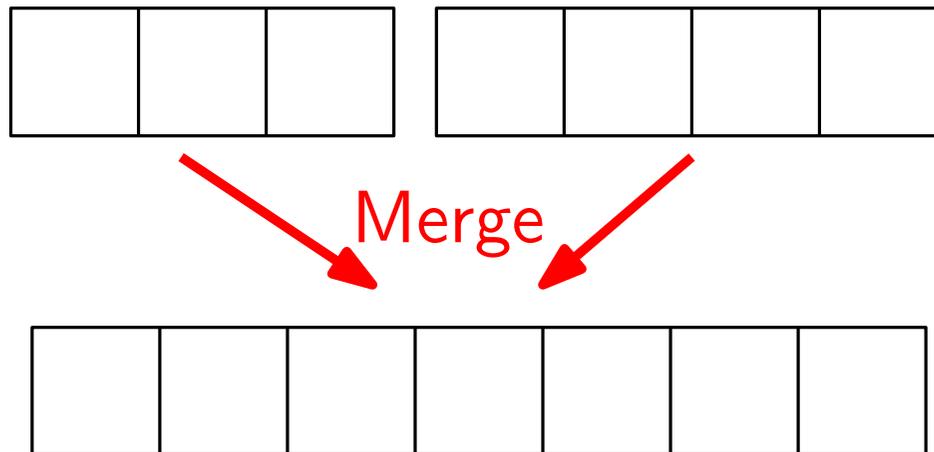
En cada paso, hubo un movimiento de flecha a la derecha en alguno de los arreglos originales. $O(n)$

Ideas previas para Merge Sort

- ▶ Un arreglo de tamaño 1 ya está ordenado.

15

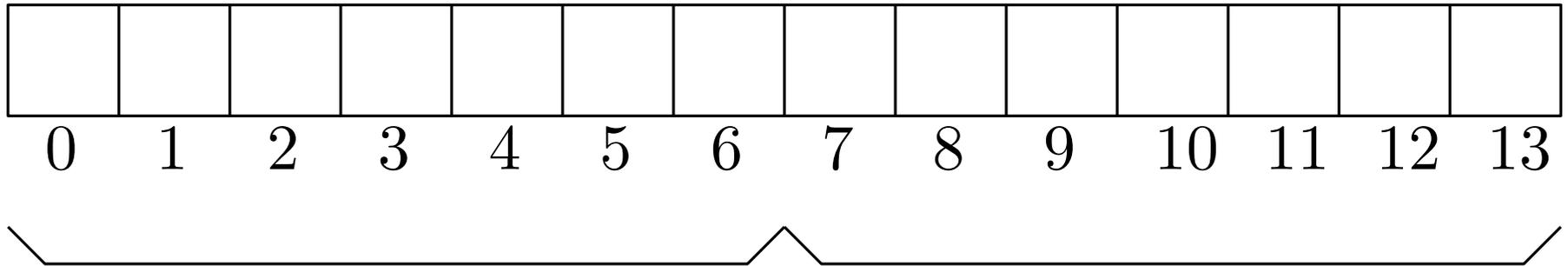
- ▶ Dos arreglos ordenados de tamaño similar se pueden mezclar (Merge).



Merge Sort

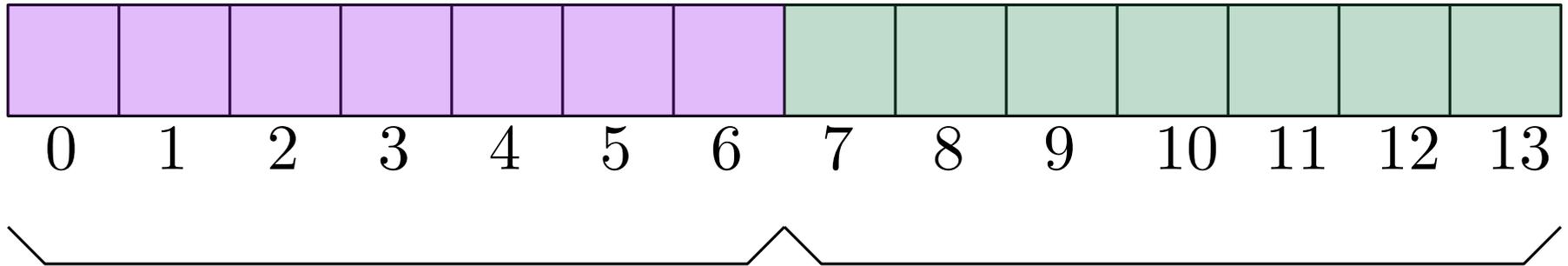
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Merge Sort



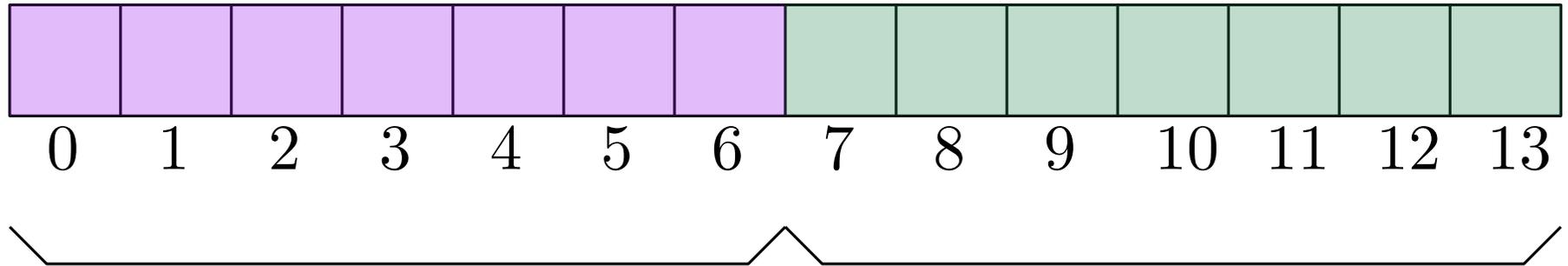
- ▶ Partir el arreglo en 2 partes del *mismo* tamaño.

Merge Sort



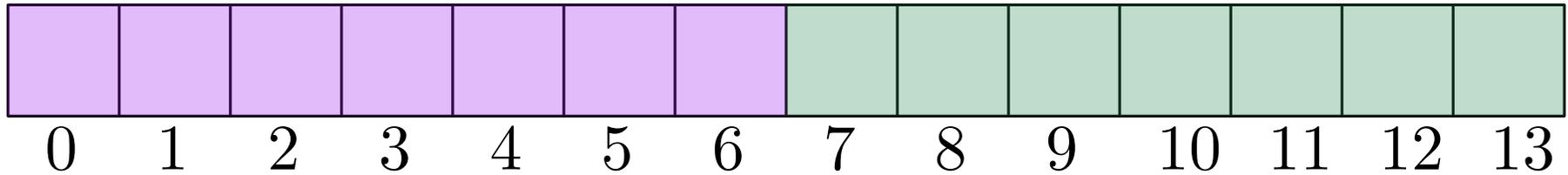
- ▶ Partir el arreglo en 2 partes del *mismo* tamaño.
- ▶ Ordena cada parte con Merge Sort (2 llamadas recursivas).

Merge Sort



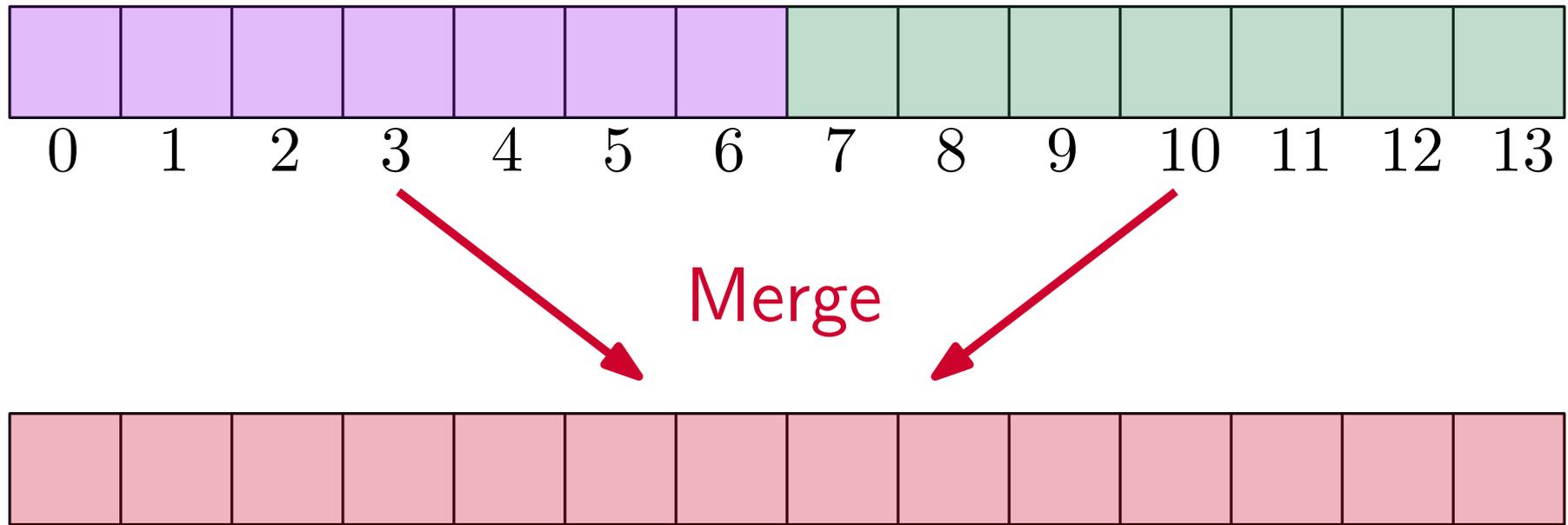
- ▶ Partir el arreglo en 2 partes del *mismo* tamaño.
- ▶ Ordena cada parte con Merge Sort (2 llamadas recursivas).
- ▶ Cuando se llega a una parte de tamaño 1, ya está ordenada.

Merge Sort



- ▶ Partir el arreglo en 2 partes del *mismo* tamaño.
- ▶ Ordena cada parte con Merge Sort (2 llamadas recursivas).
- ▶ Cuando se llega a una parte de tamaño 1, ya está ordenada.
- ▶ Cuando tengas 2 partes ya ordenadas Mézclalas (Merge).

Merge Sort



- ▶ Partir el arreglo en 2 partes del *mismo* tamaño.
- ▶ Ordena cada parte con Merge Sort (2 llamadas recursivas).
- ▶ Cuando se llega a una parte de tamaño 1, ya está ordenada.
- ▶ Cuando tengas 2 partes ya ordenadas Mézclalas (Merge).

Merge Sort

15	7	-1	5	2	8	-1	8	3	-6	6	9	31	17
----	---	----	---	---	---	----	---	---	----	---	---	----	----

Merge Sort

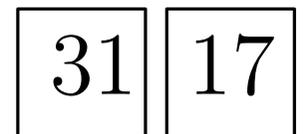
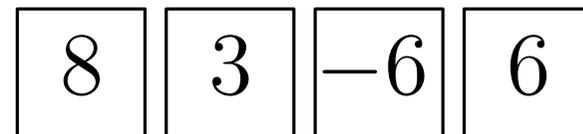
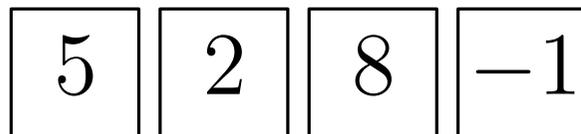
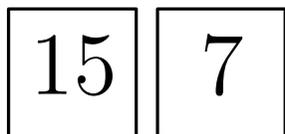
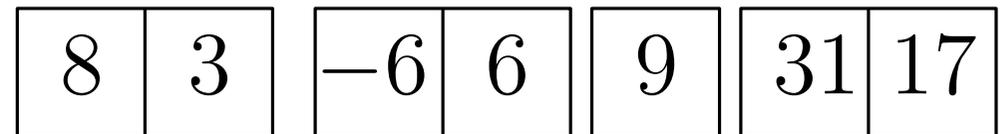
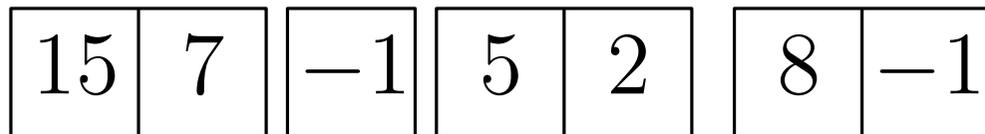
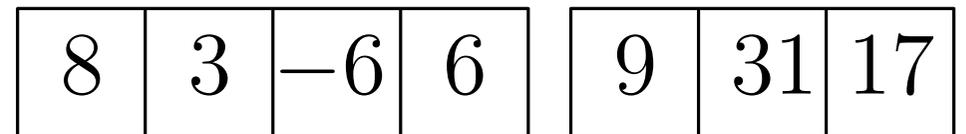
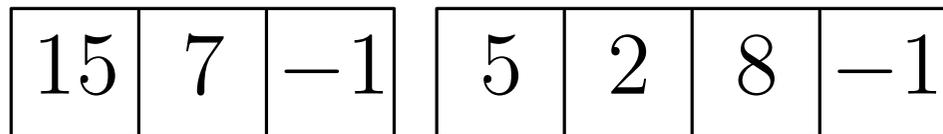
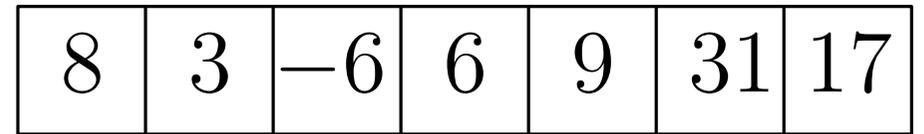
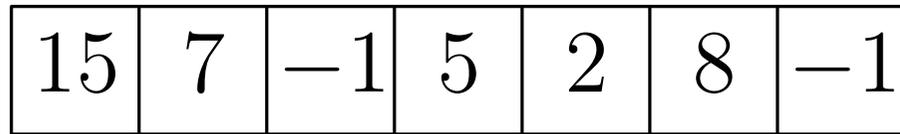
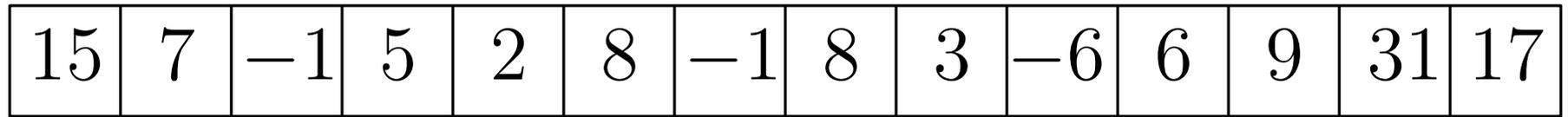
15	7	-1	5	2	8	-1	8	3	-6	6	9	31	17
----	---	----	---	---	---	----	---	---	----	---	---	----	----



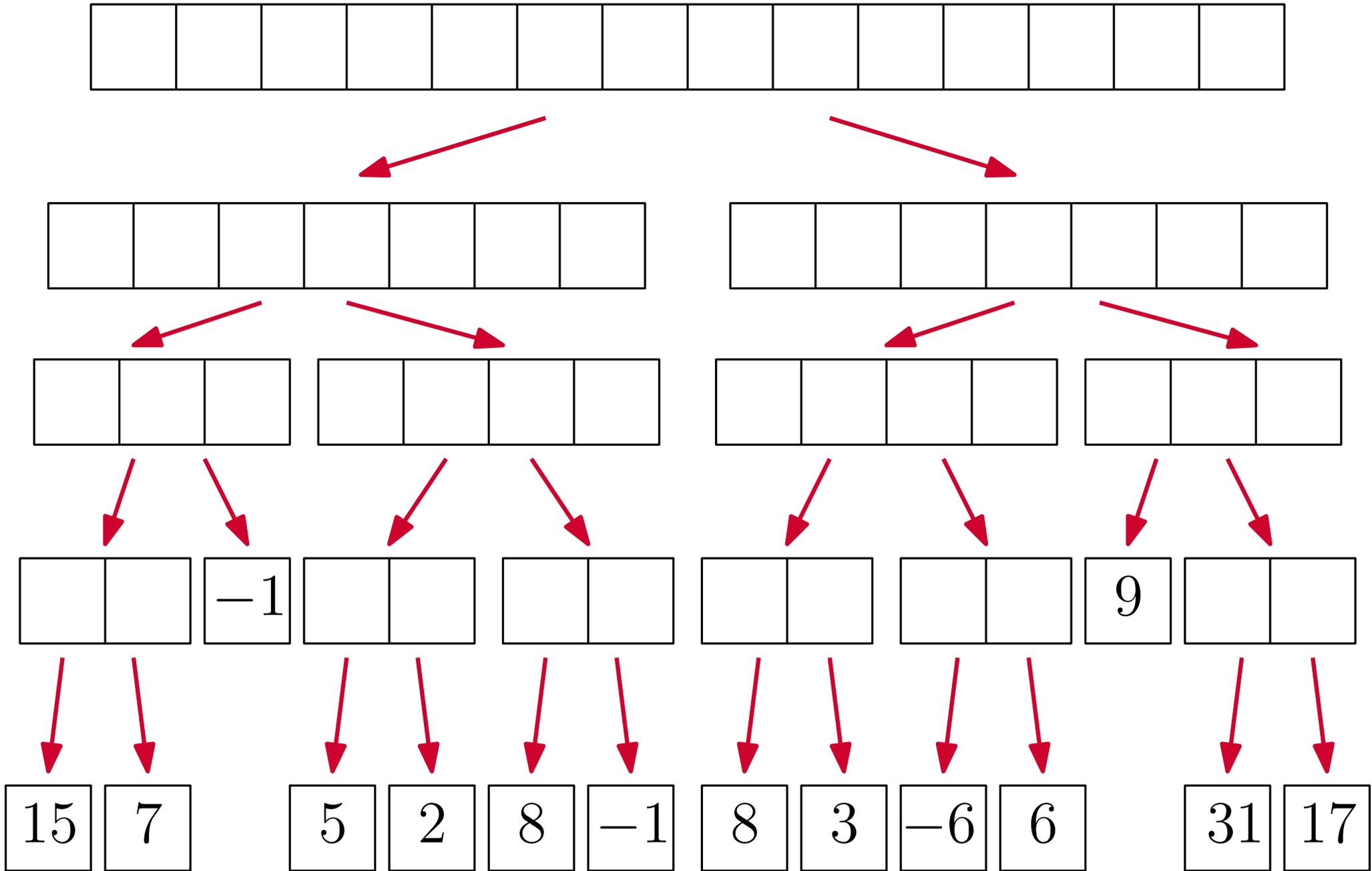
15	7	-1	5	2	8	-1
----	---	----	---	---	---	----

8	3	-6	6	9	31	17
---	---	----	---	---	----	----

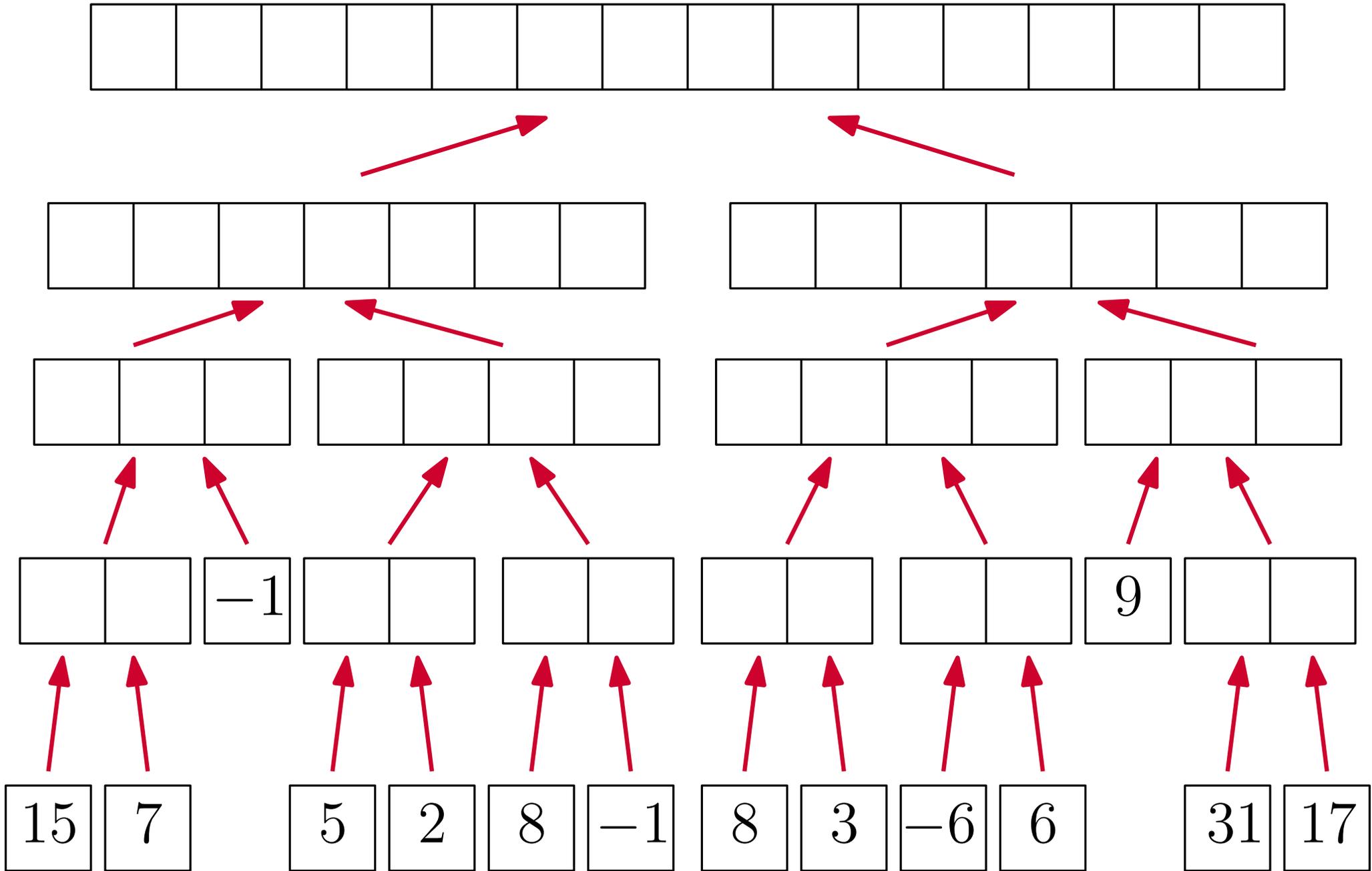
Merge Sort



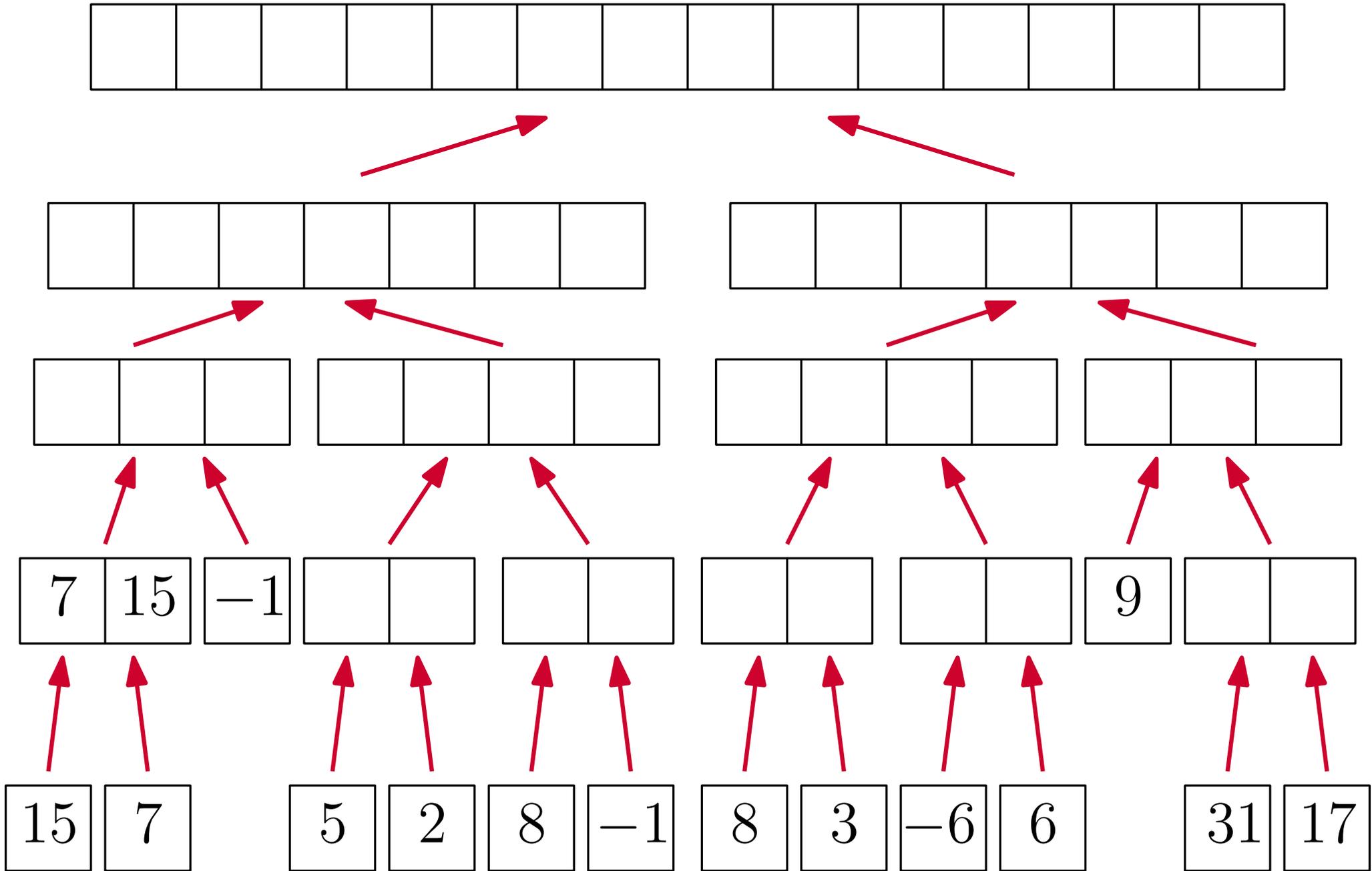
Merge Sort



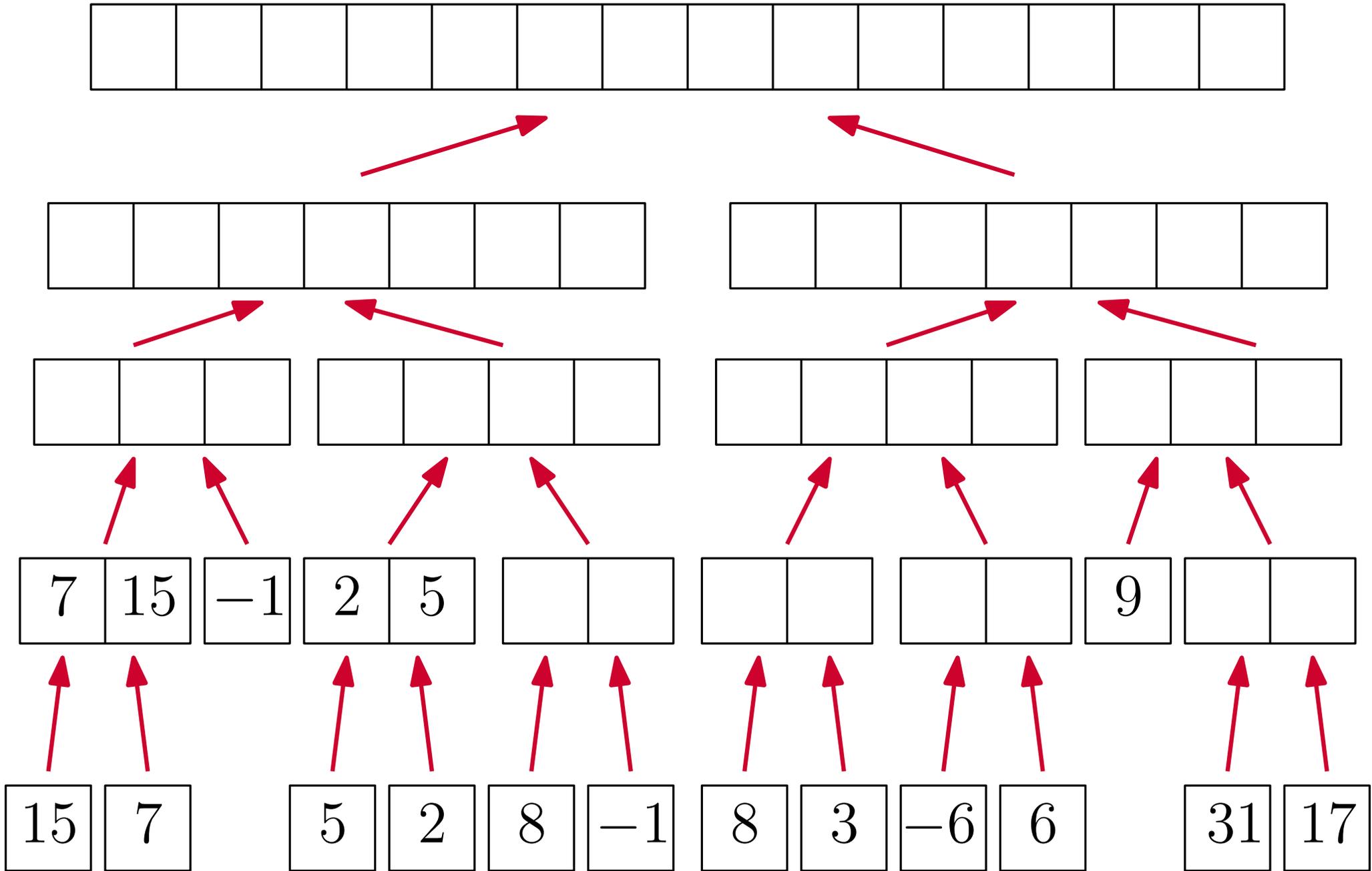
Merge Sort



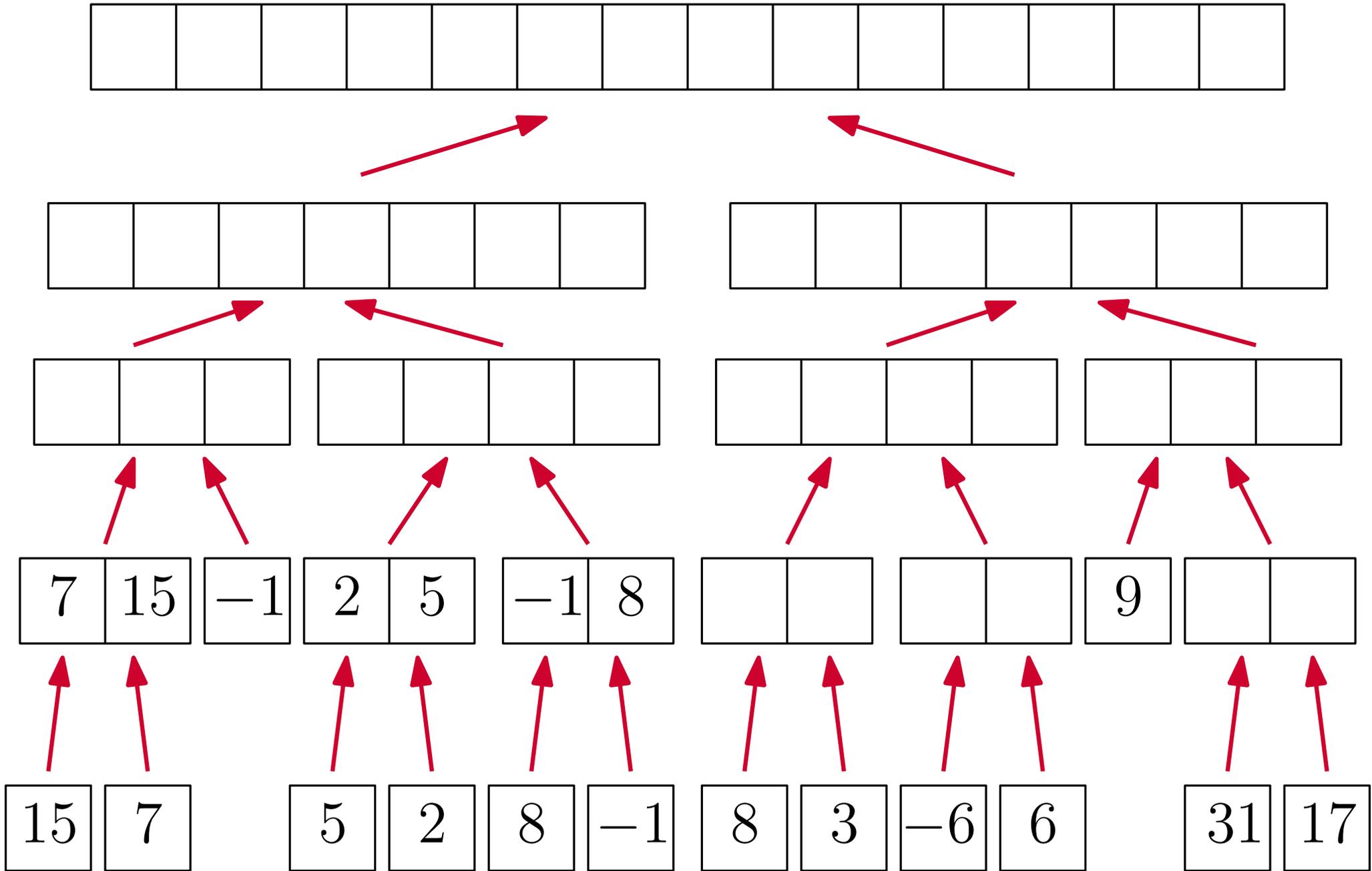
Merge Sort



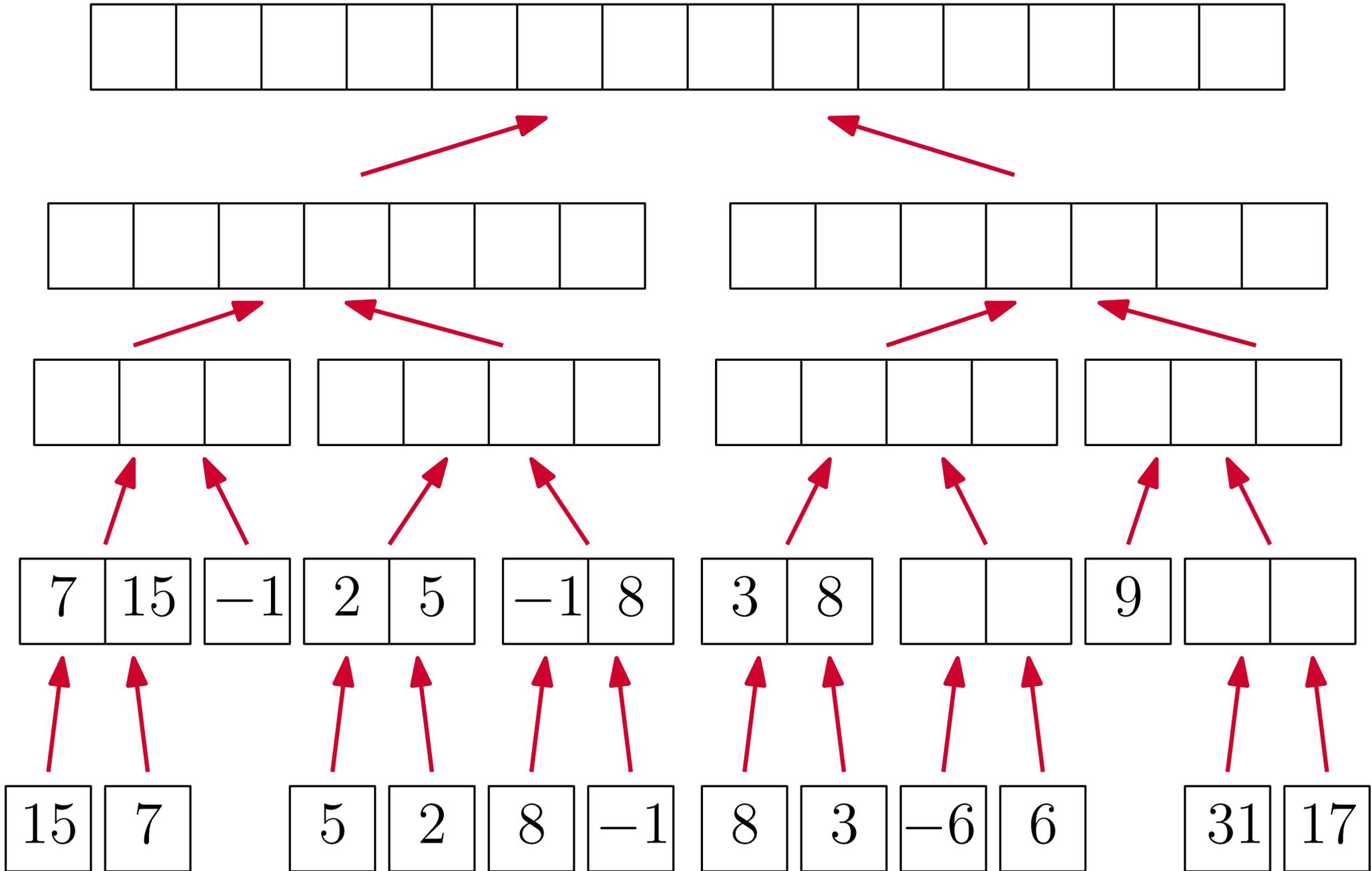
Merge Sort



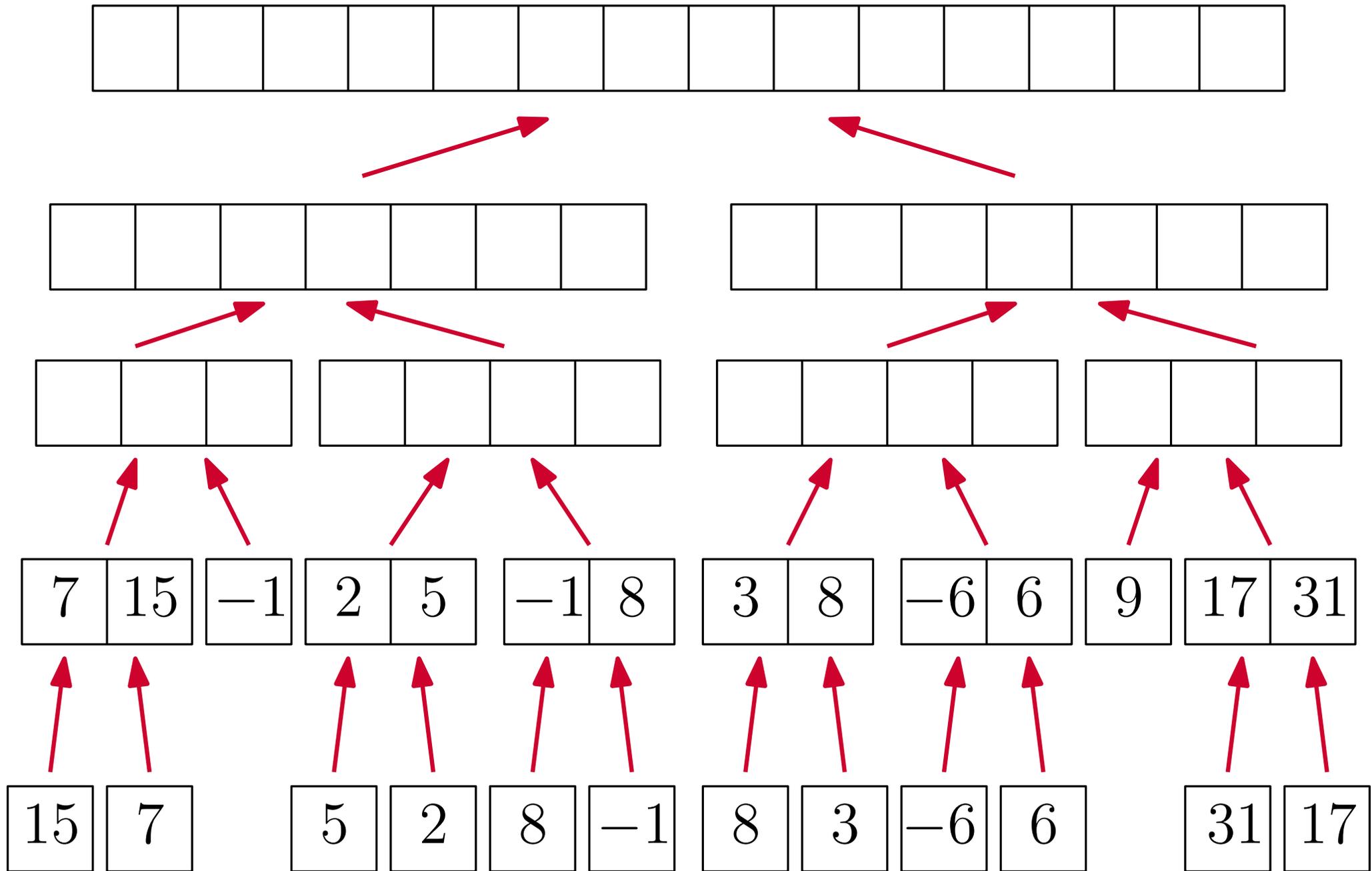
Merge Sort



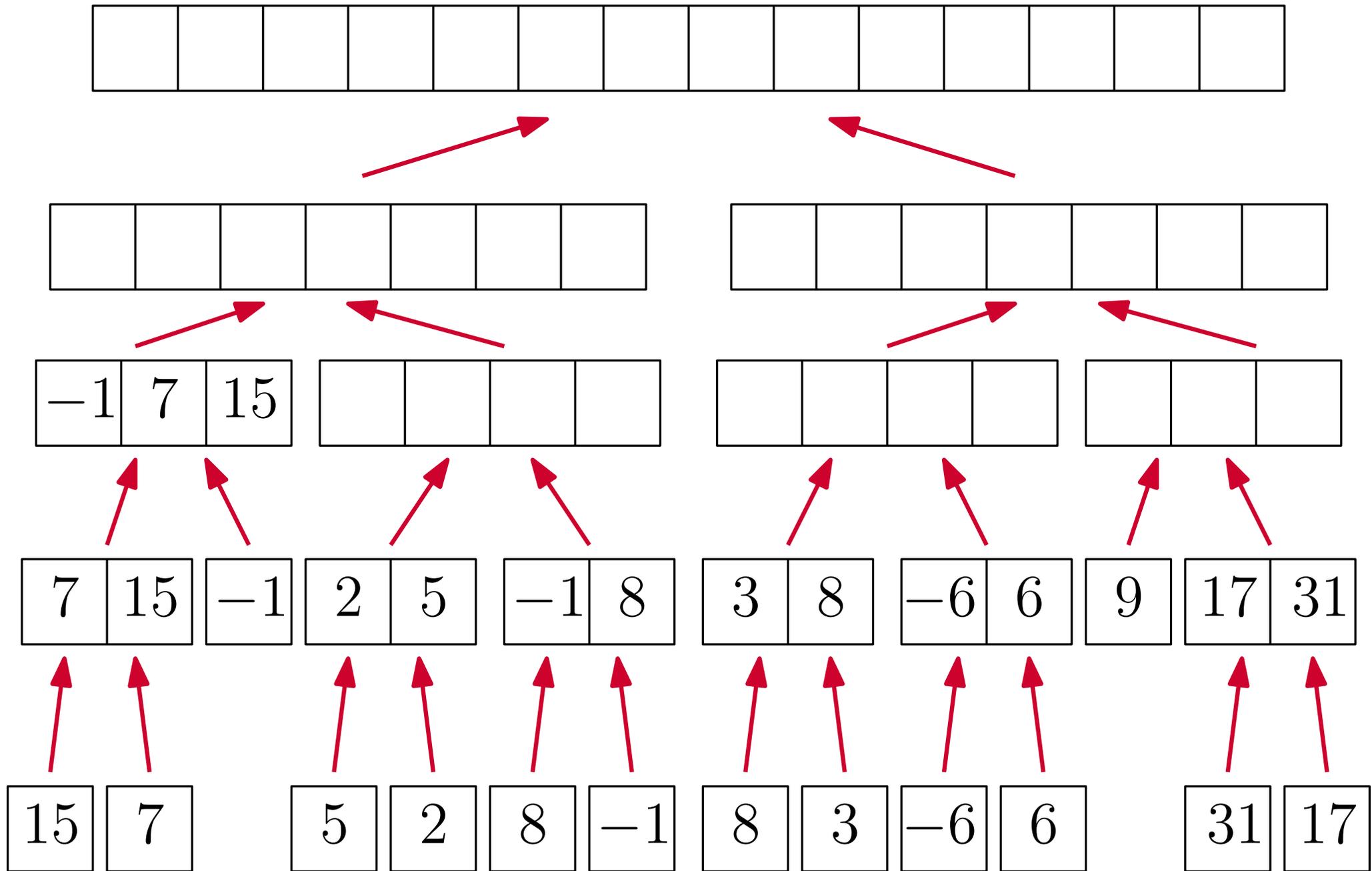
Merge Sort



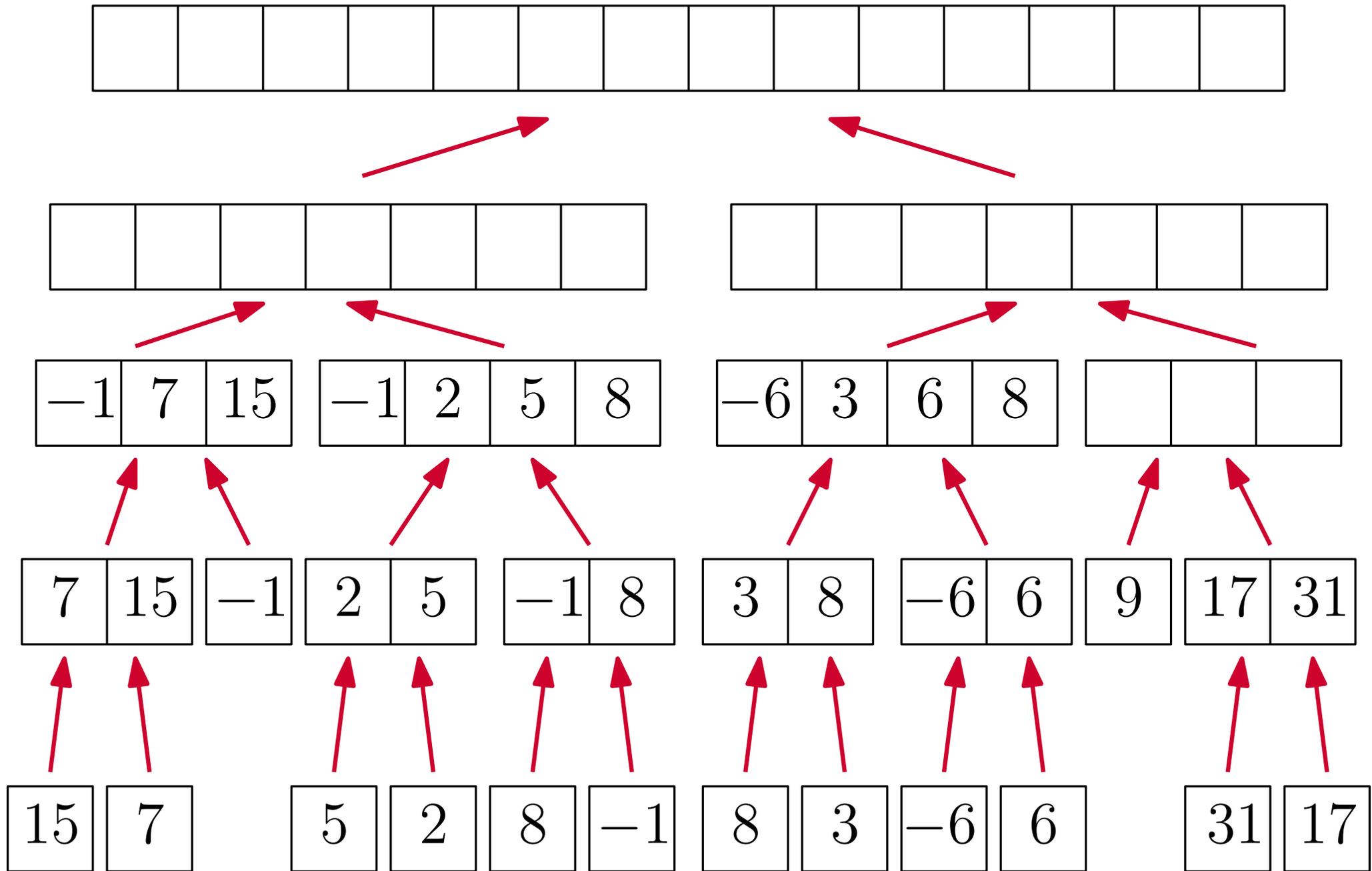
Merge Sort



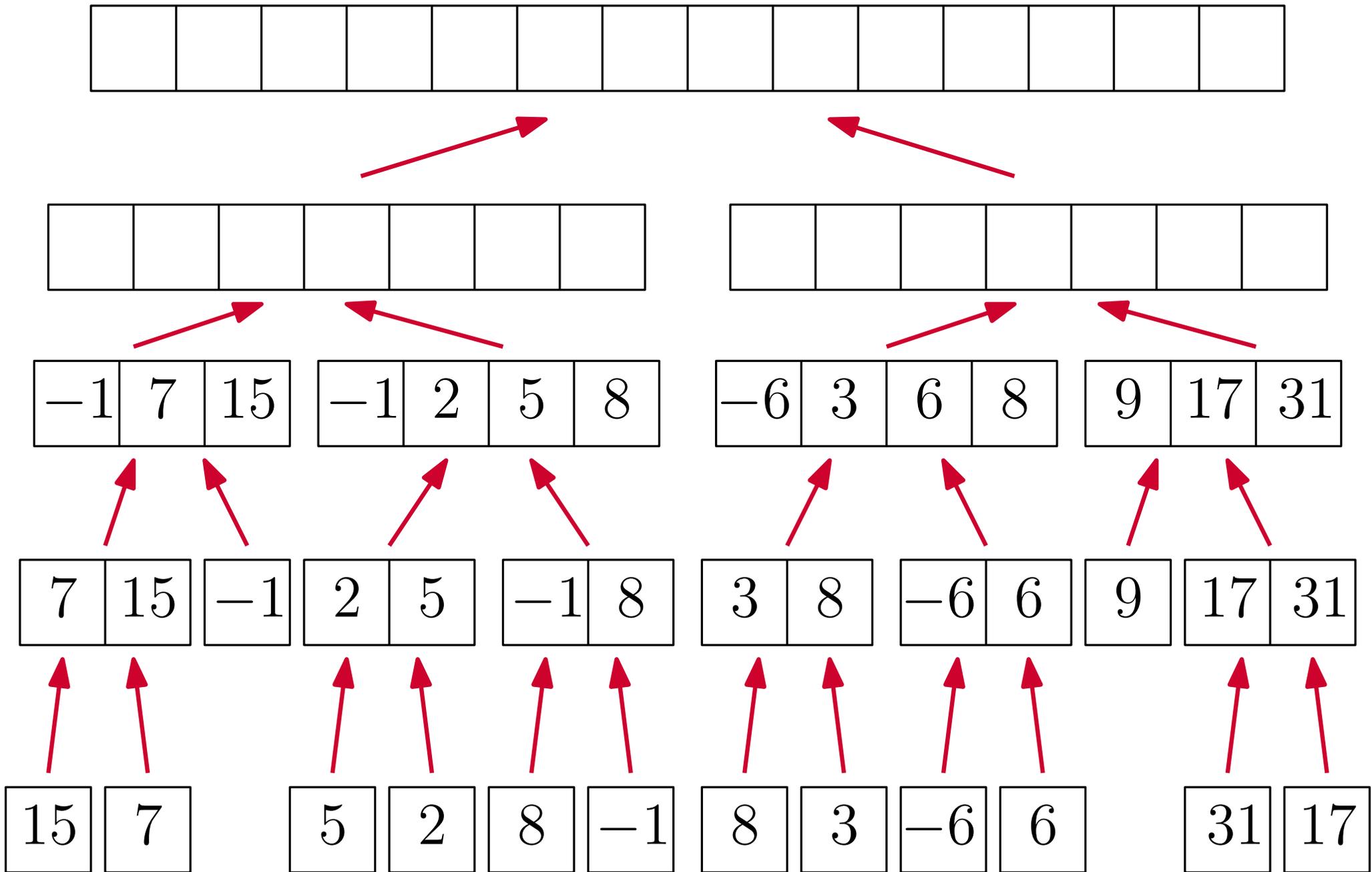
Merge Sort



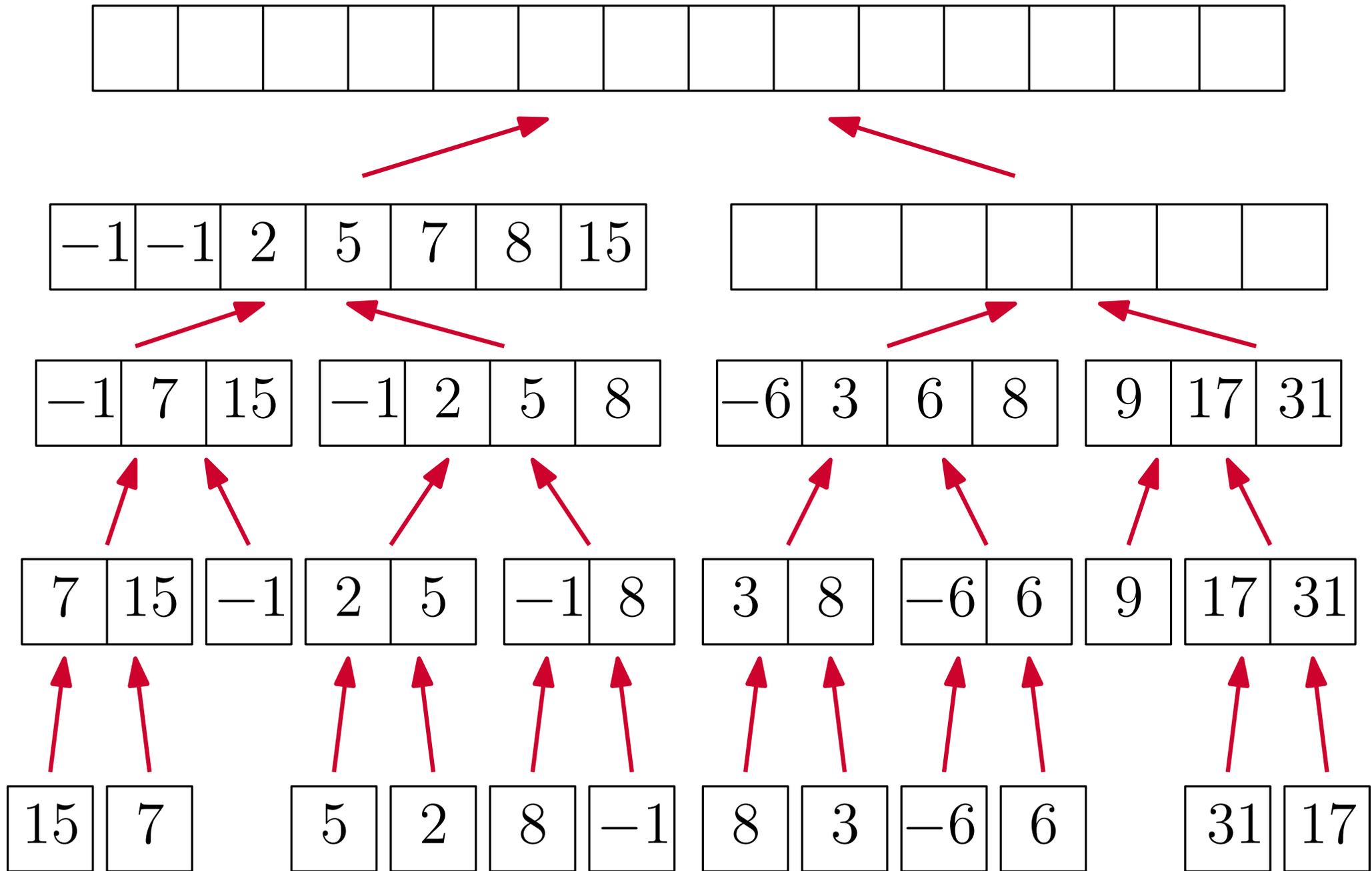
Merge Sort



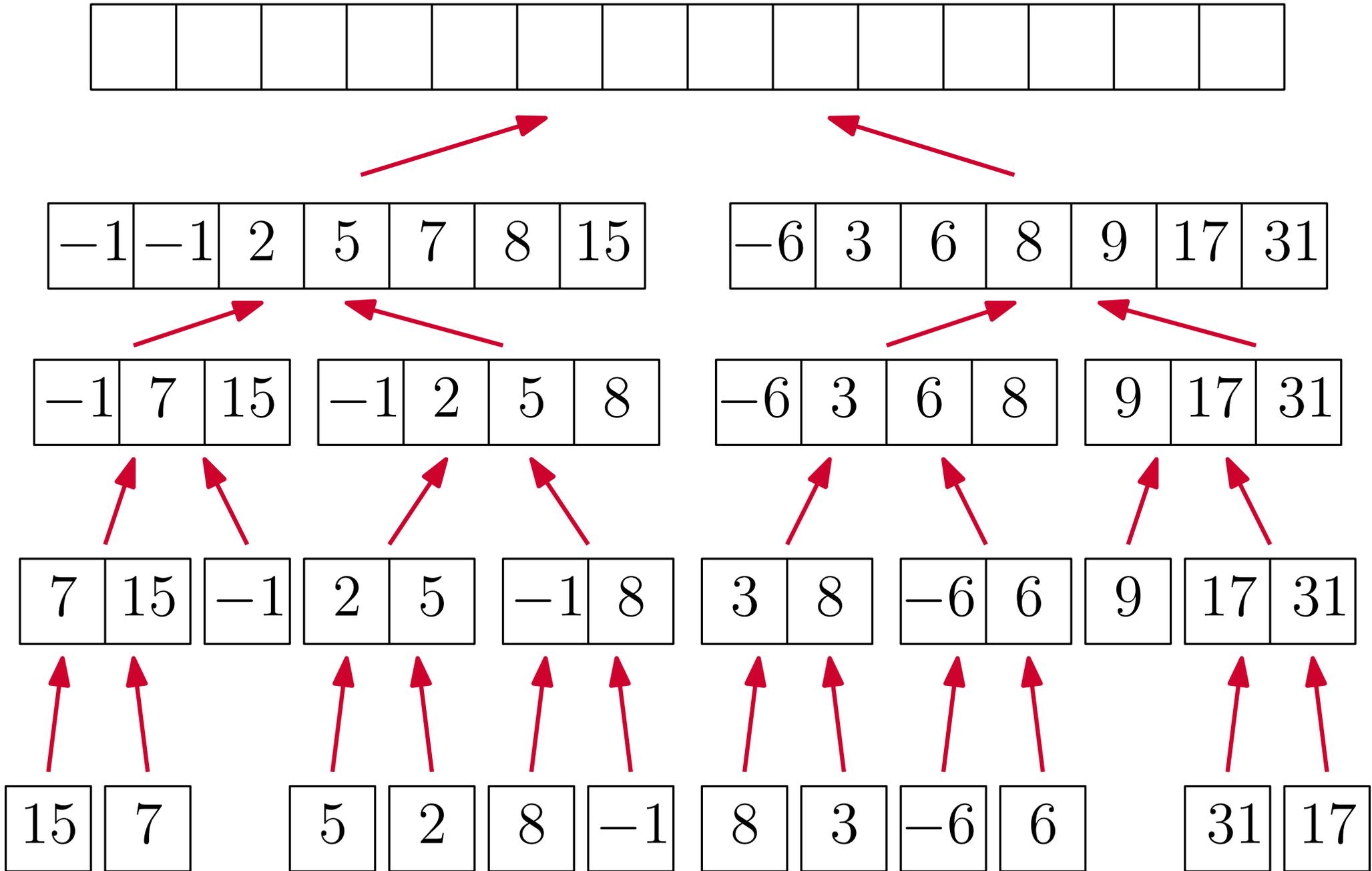
Merge Sort



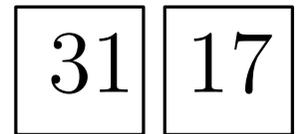
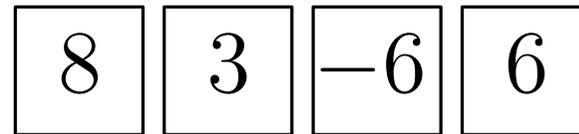
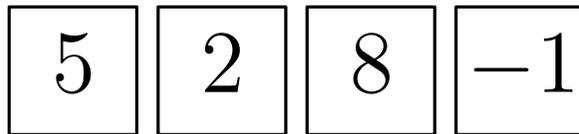
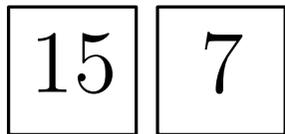
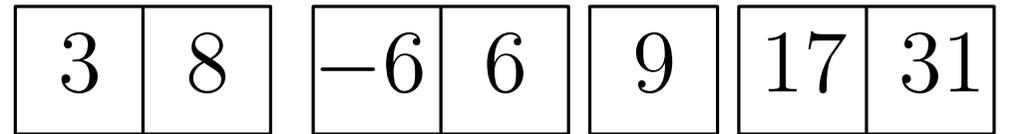
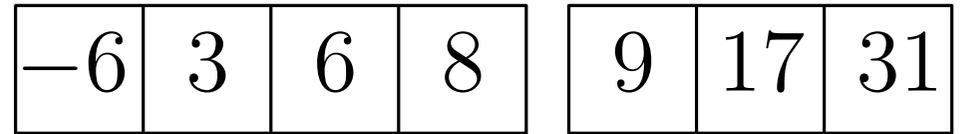
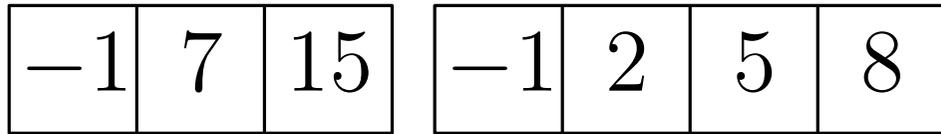
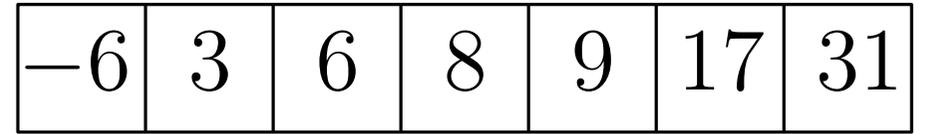
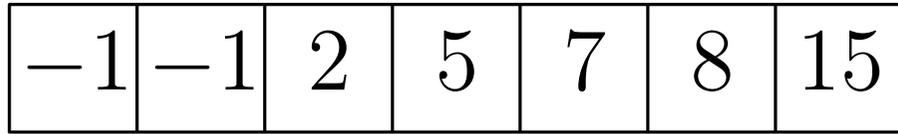
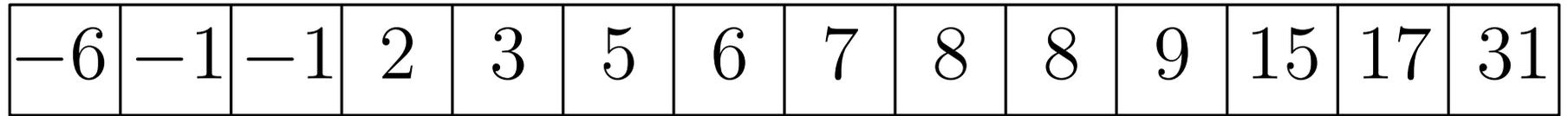
Merge Sort



Merge Sort



Merge Sort



Complejidad Merge Sort

Tamaño
arreglo

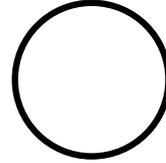
Piezas

Complejidad Merge Sort

Tamaño
arreglo

n

Piezas



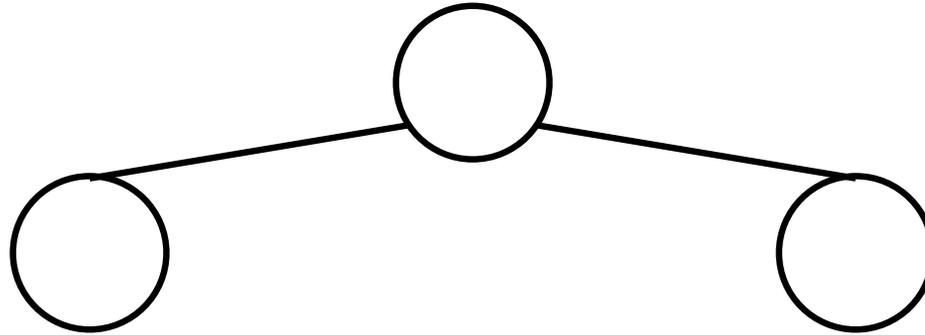
Complejidad Merge Sort

Tamaño
arreglo

n

$\frac{n}{2}$

Piezas



Complejidad Merge Sort

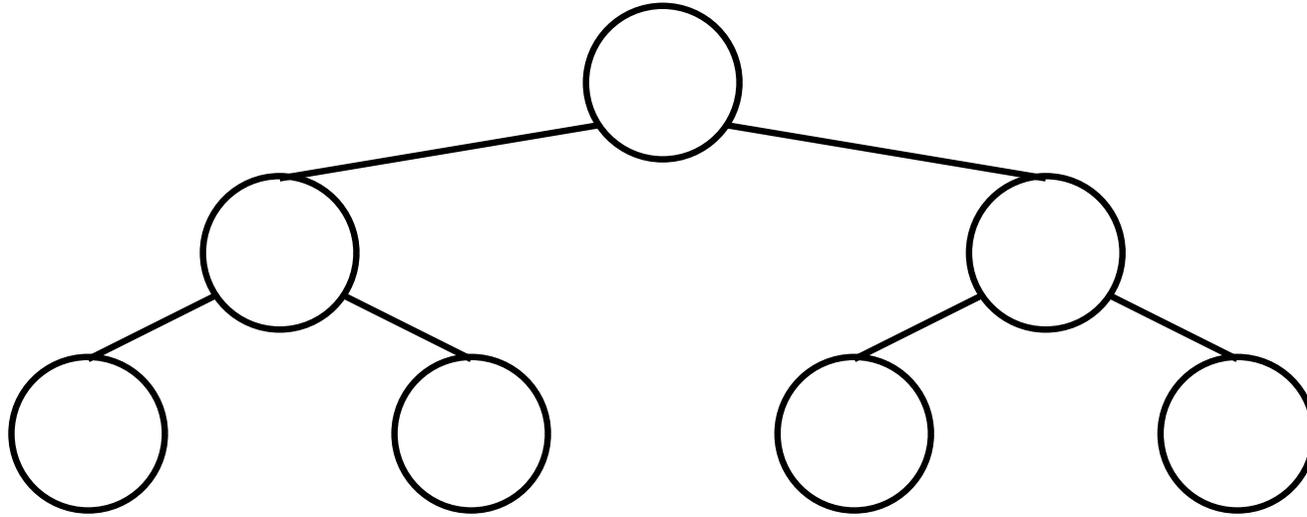
Tamaño
arreglo

Piezas

n

$\frac{n}{2}$

$\frac{n}{4}$



Complejidad Merge Sort

Tamaño
arreglo

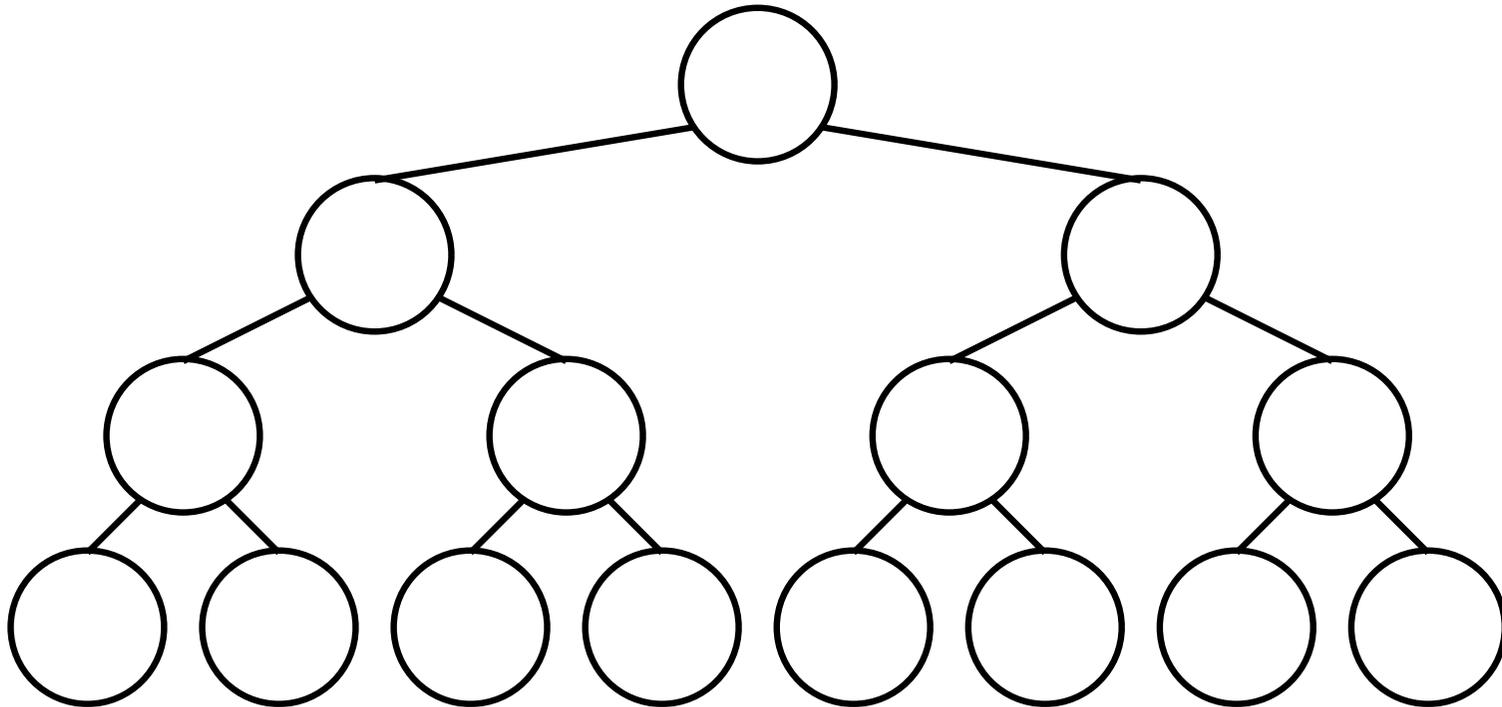
Piezas

n

$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$



Complejidad Merge Sort

Tamaño
arreglo

Piezas

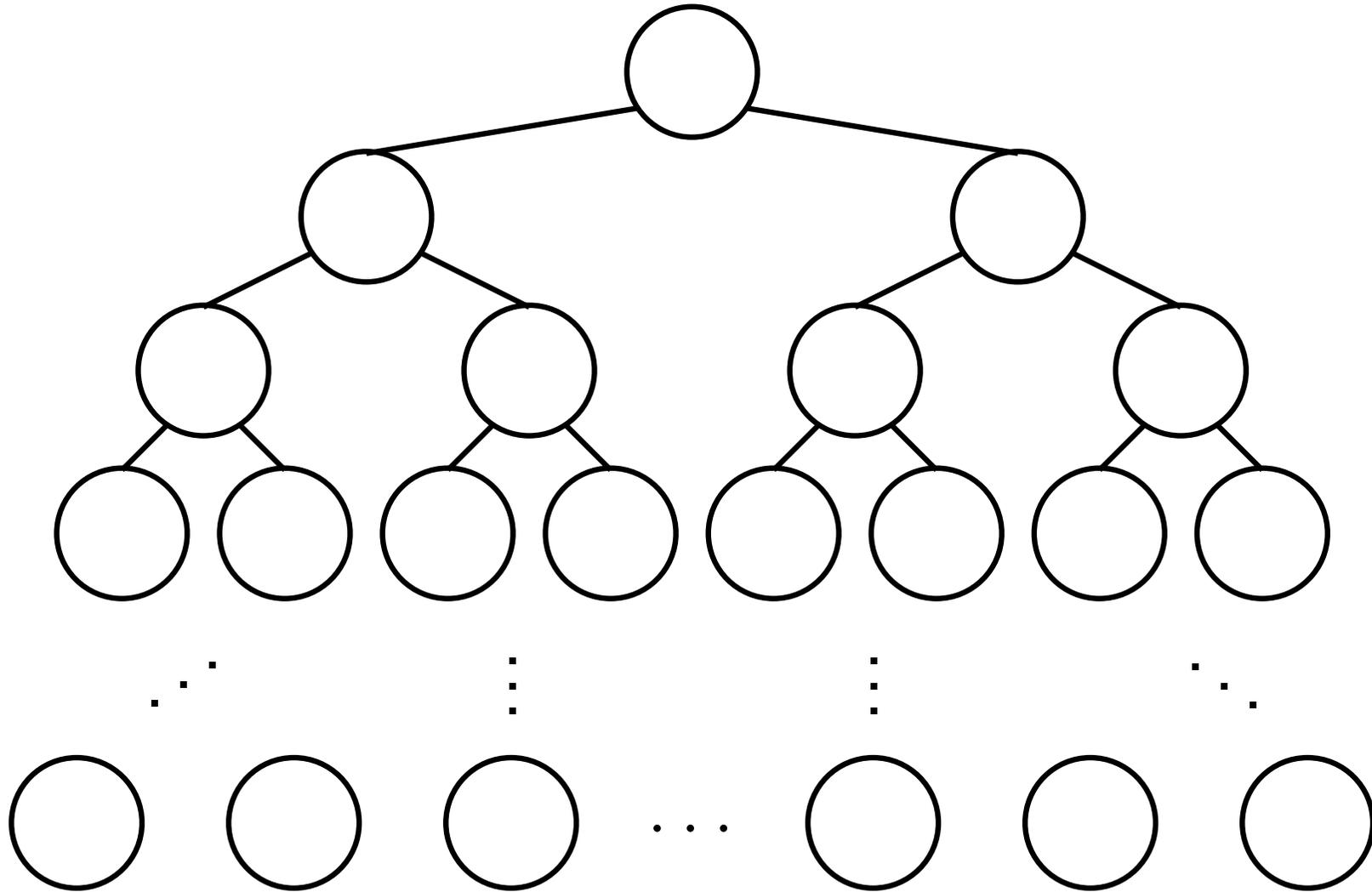
n

$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$

2



Complejidad Merge Sort

Tamaño arreglo

Piezas

n

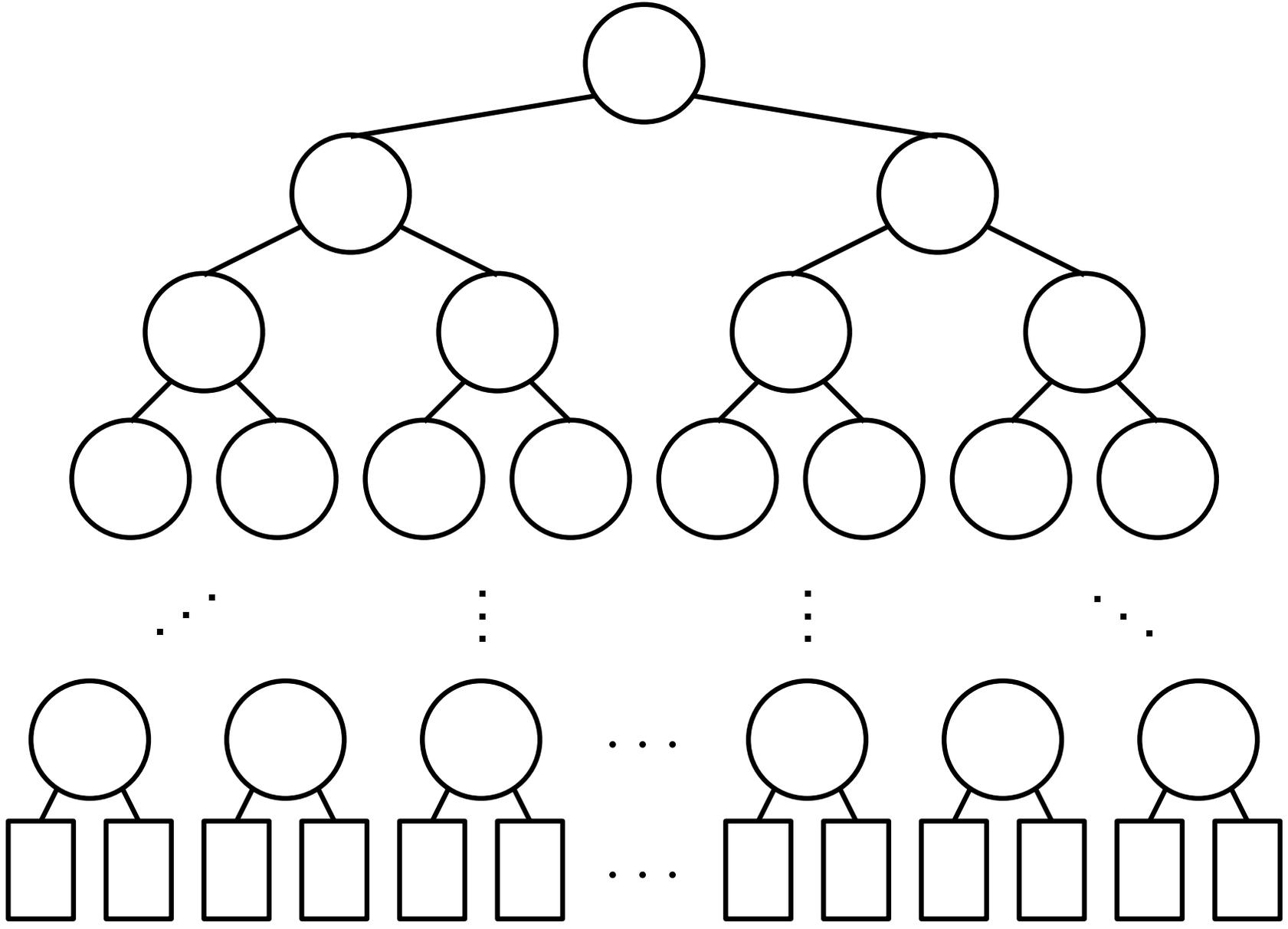
$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$

2

1



Complejidad Merge Sort

Tamaño arreglo

Piezas

¿# de operaciones?

n

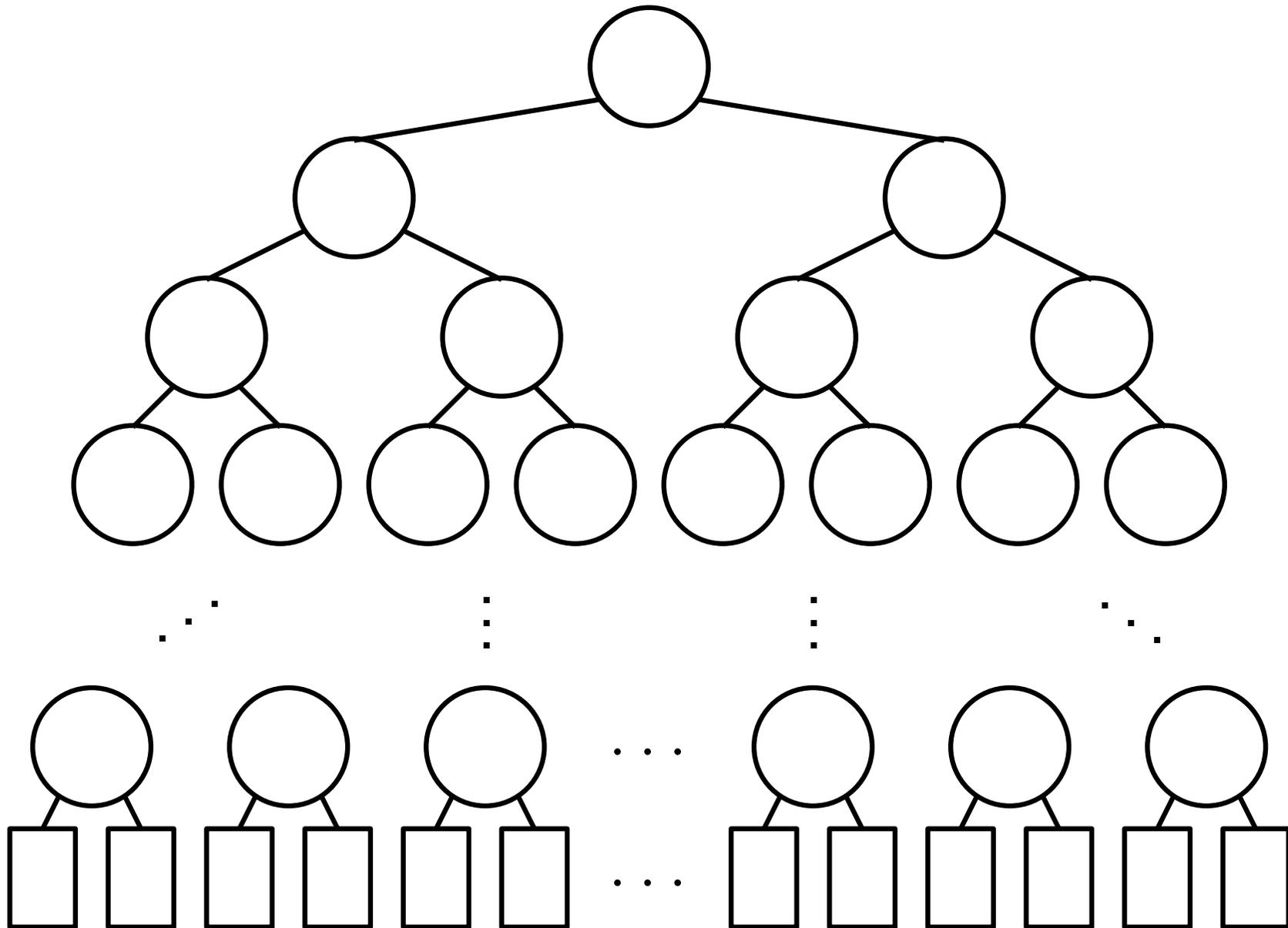
$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$

2

1



Complejidad Merge Sort

Tamaño arreglo

Piezas

¿# de operaciones?

n

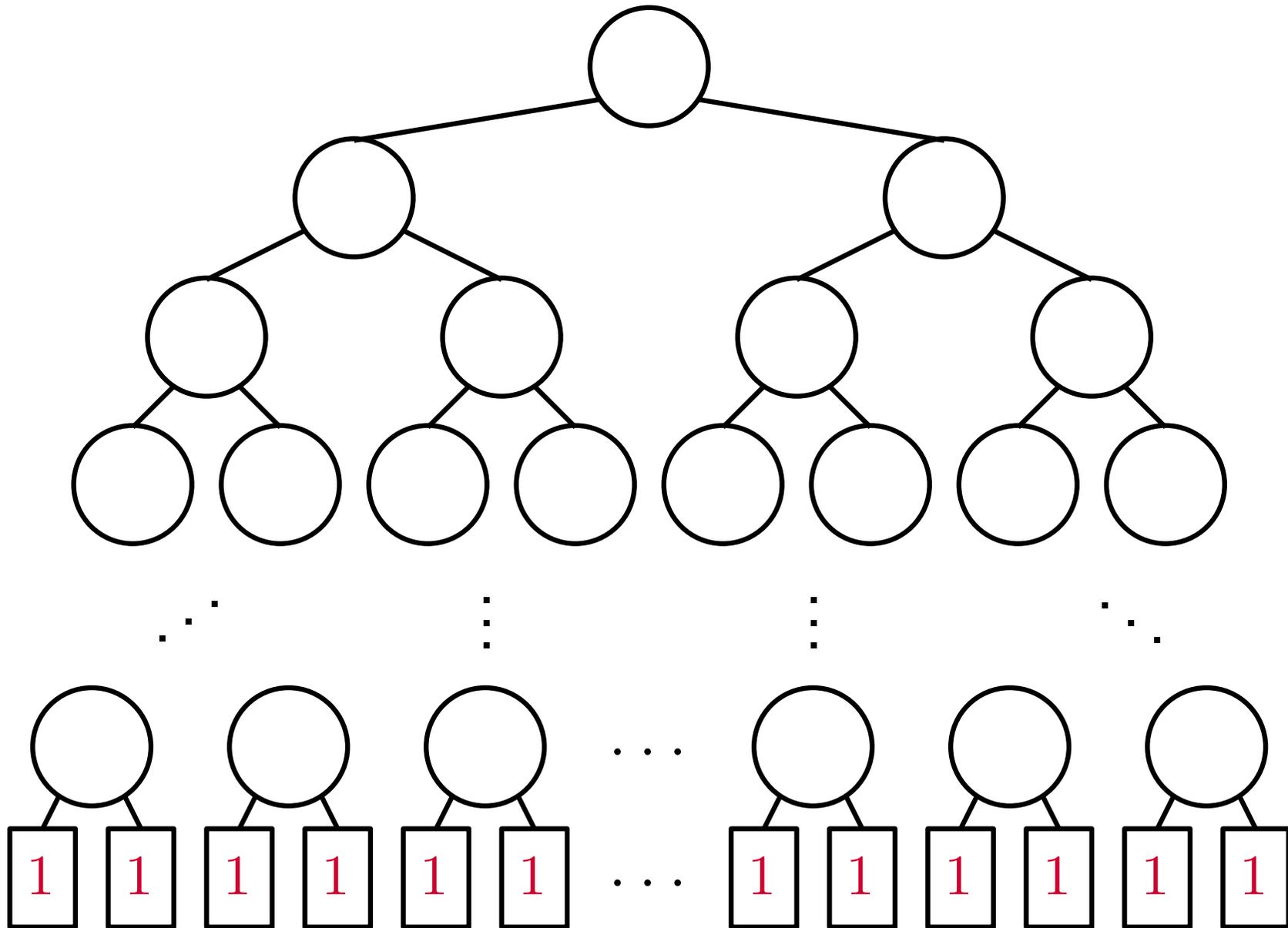
$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$

2

1



Complejidad Merge Sort

Tamaño arreglo

Piezas

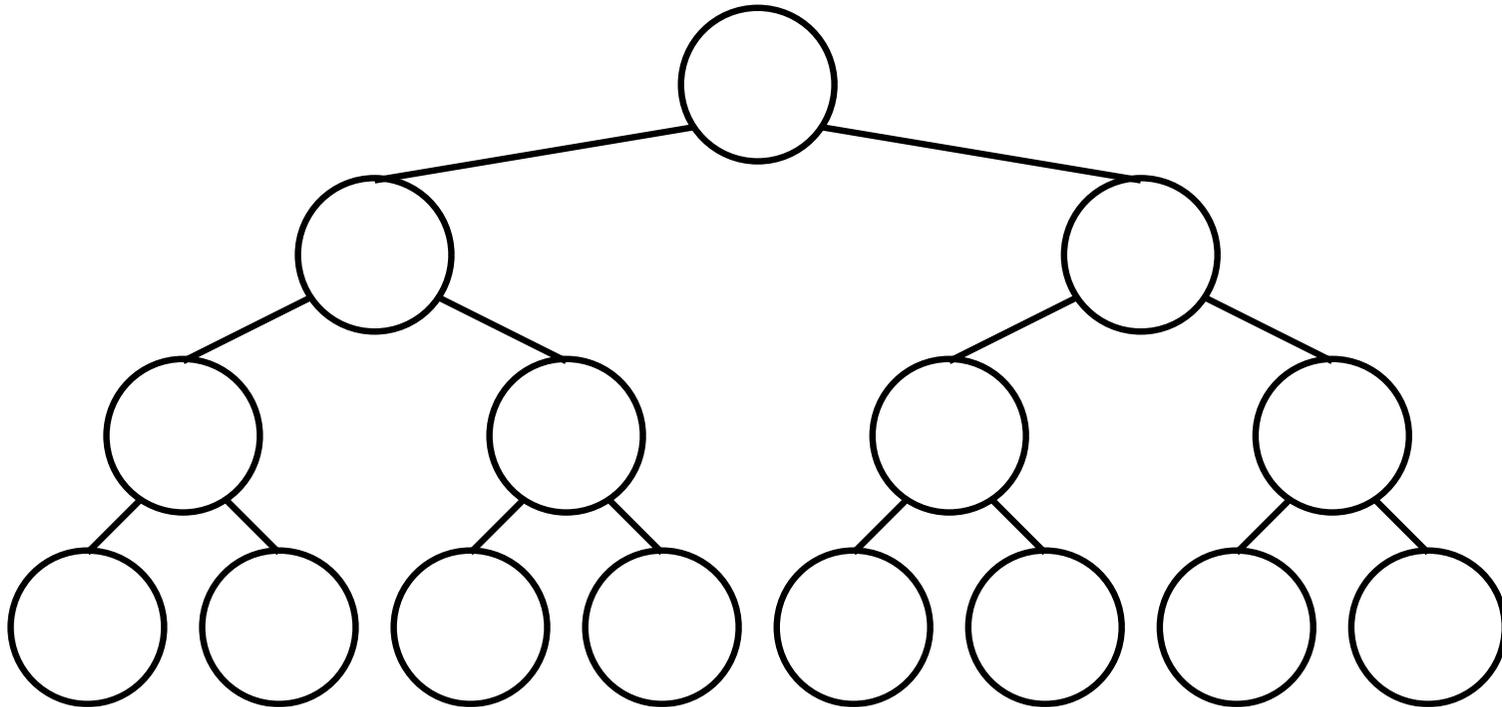
¿# de operaciones?

n

$\frac{n}{2}$

$\frac{n}{4}$

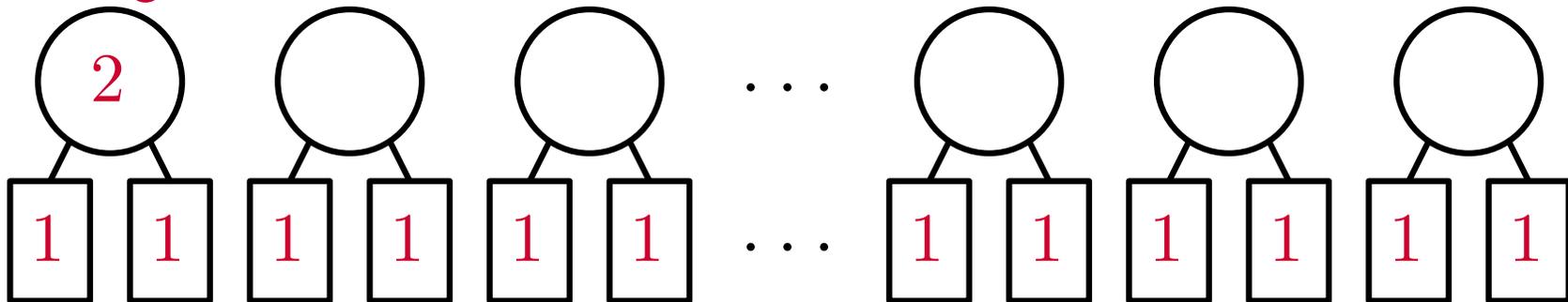
$\frac{n}{8}$



Merge

2

1



Complejidad Merge Sort

Tamaño arreglo

Piezas

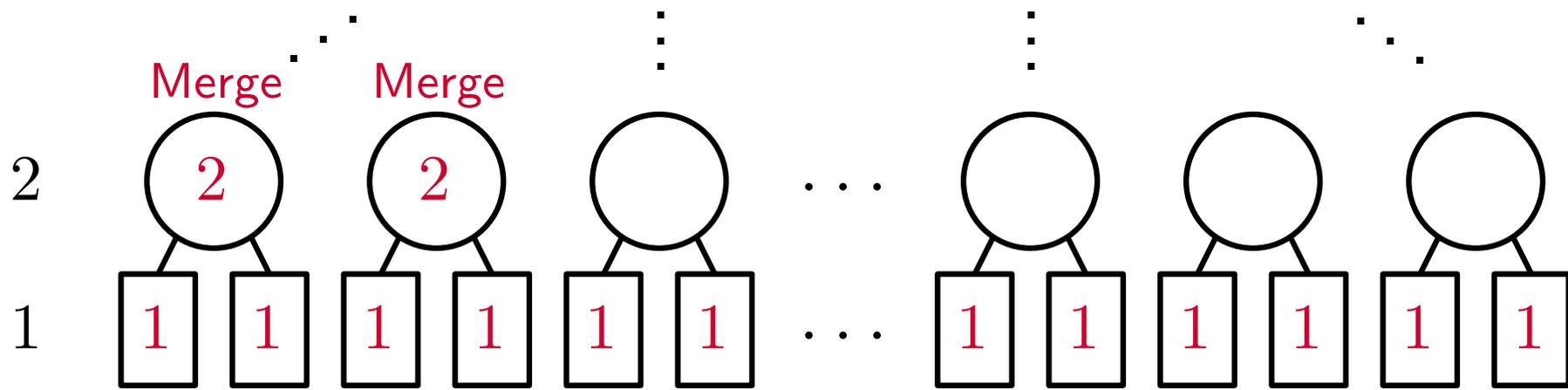
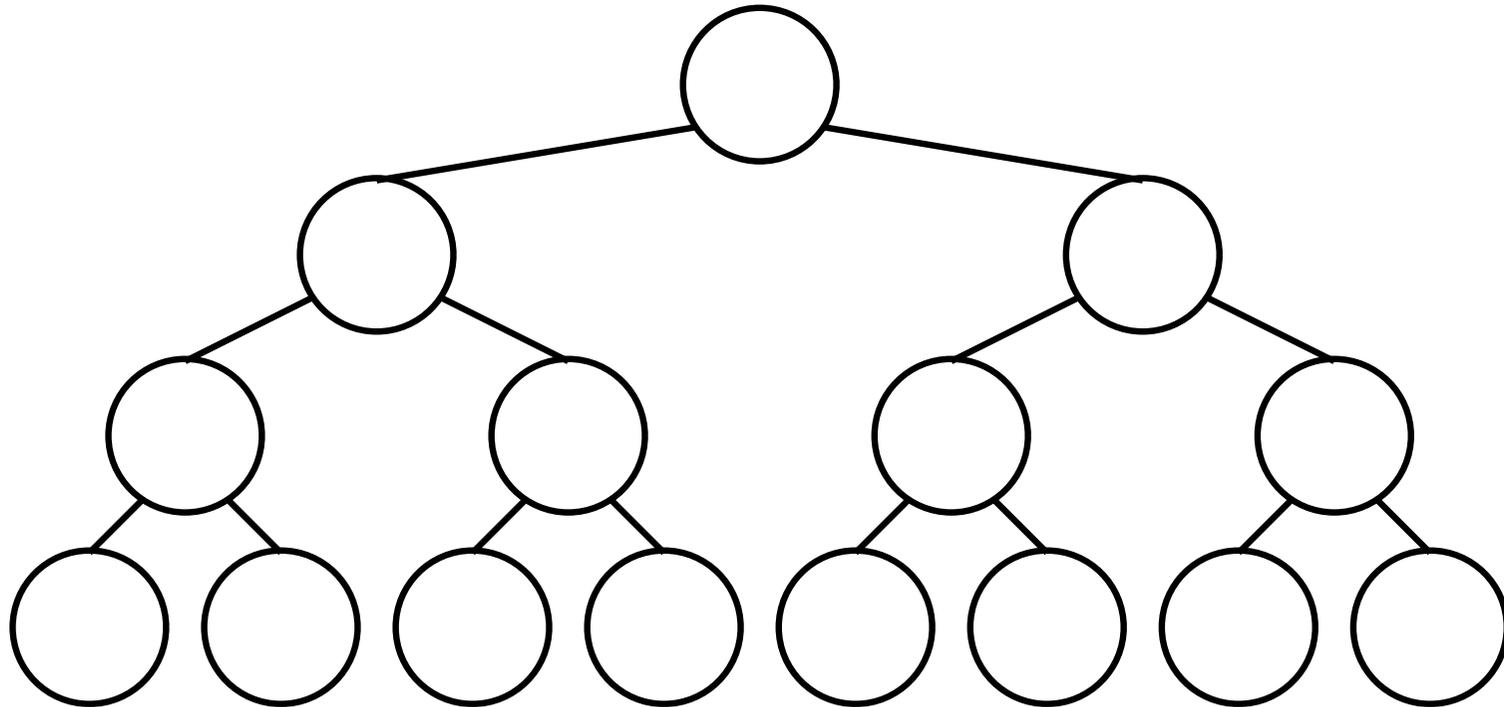
¿# de operaciones?

n

$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$



Complejidad Merge Sort

Tamaño arreglo

Piezas

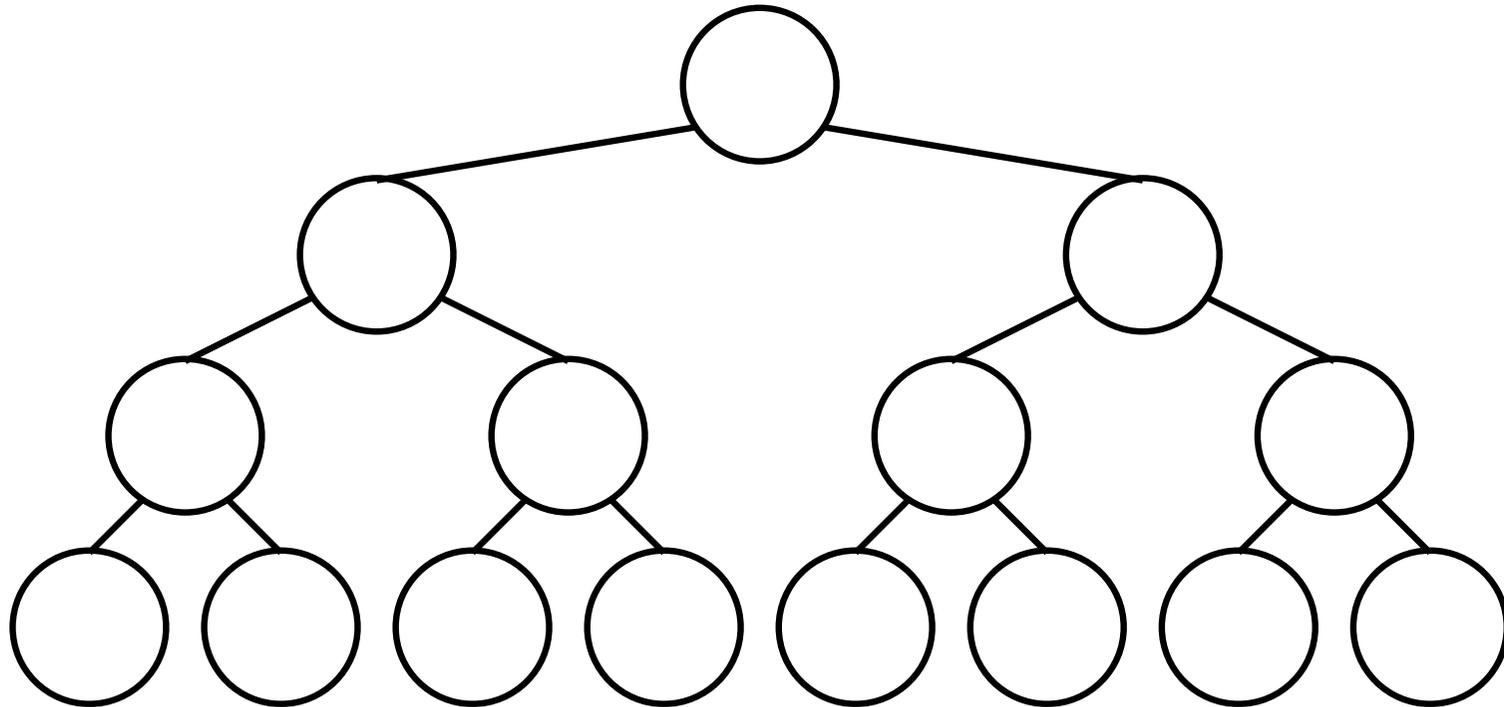
¿# de operaciones?

n

$\frac{n}{2}$

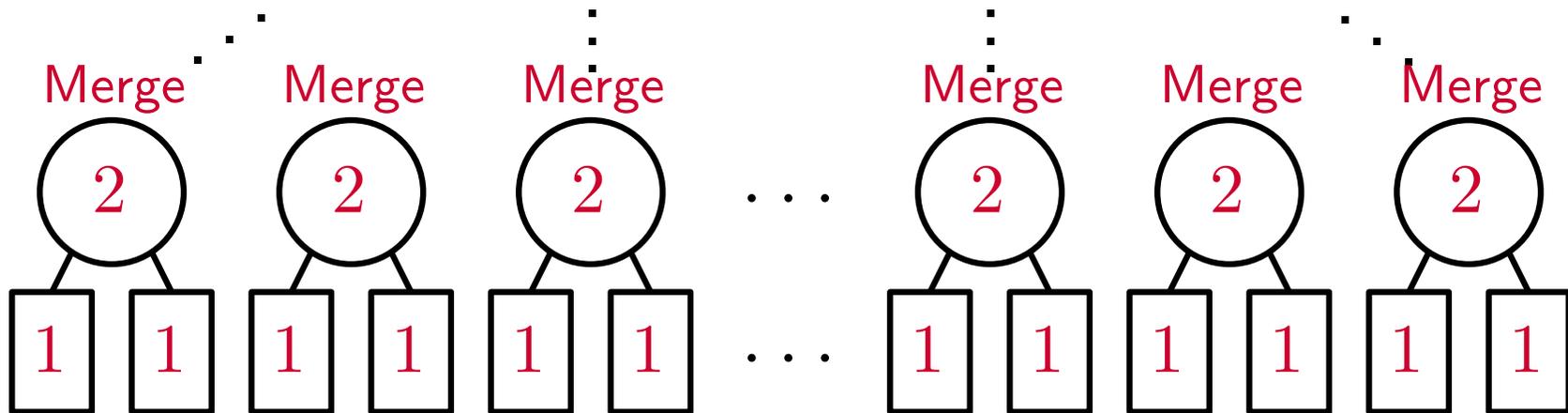
$\frac{n}{4}$

$\frac{n}{8}$



2

1



Complejidad Merge Sort

Tamaño arreglo

Piezas

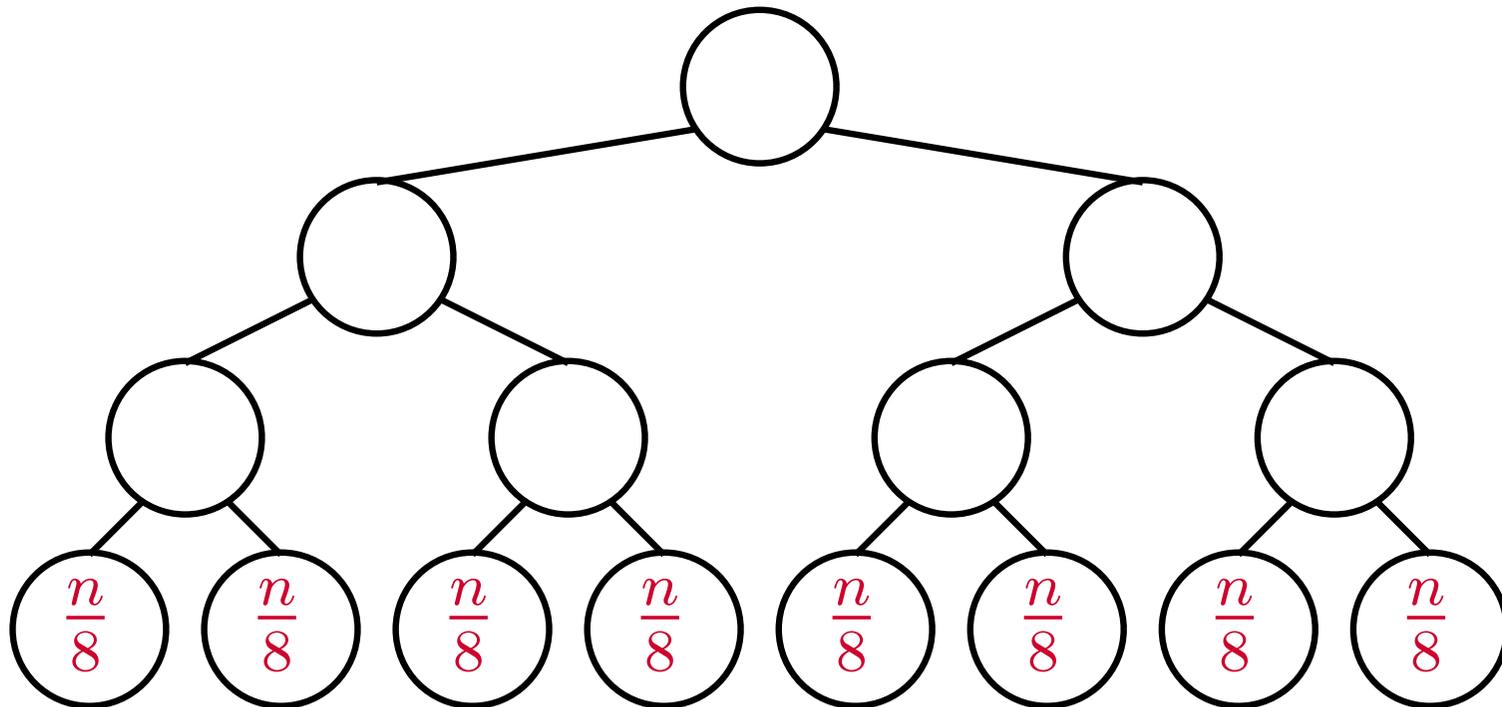
¿# de operaciones?

n

$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$



...

⋮

⋮

...

2

2

2

2

...

2

2

2

1

1

1

1

1

1

1

...

1

1

1

1

1

1

Complejidad Merge Sort

Tamaño arreglo

Piezas

¿# de operaciones?

n

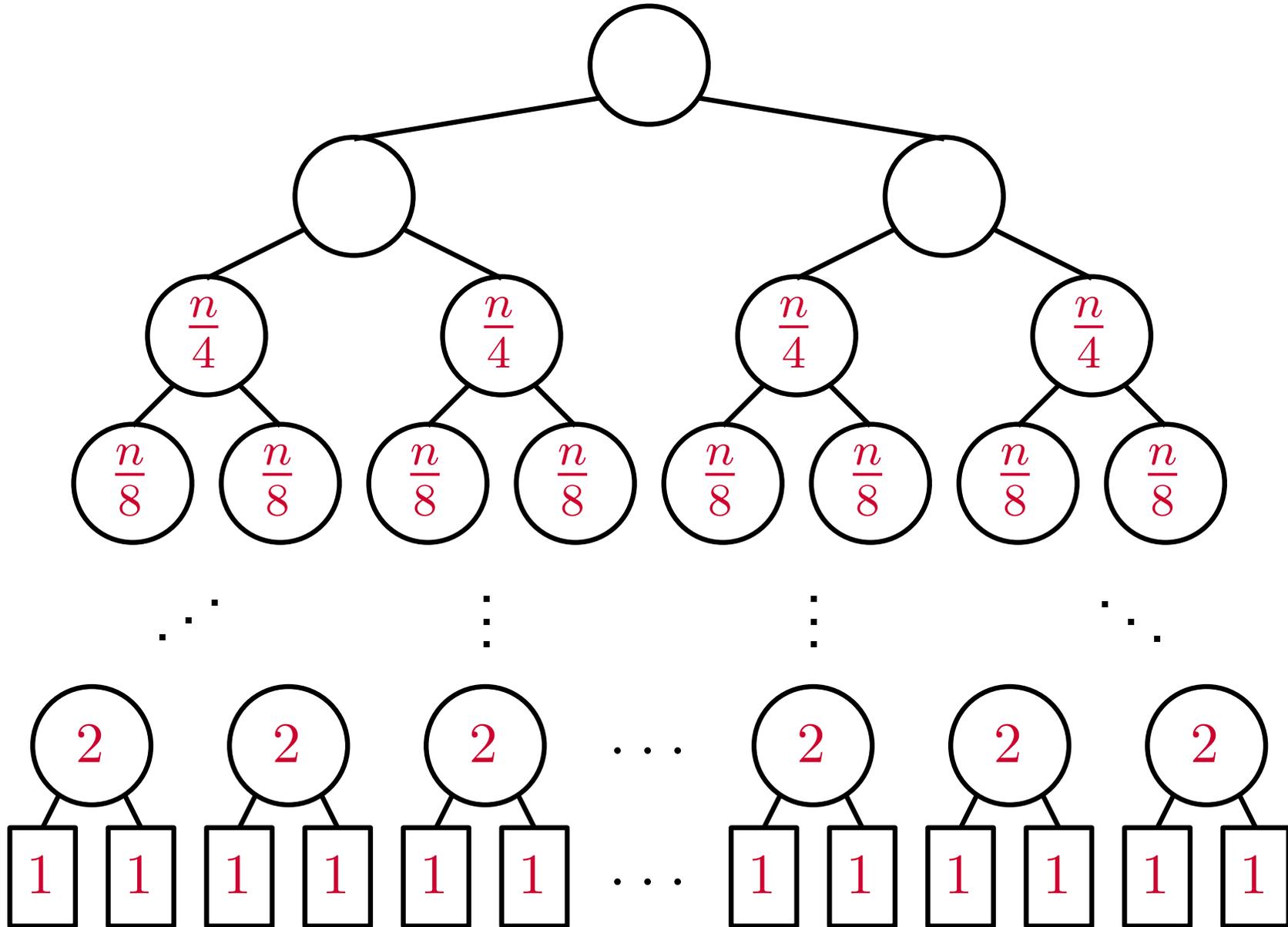
$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$

2

1



Complejidad Merge Sort

Tamaño arreglo

Piezas

¿# de operaciones?

n

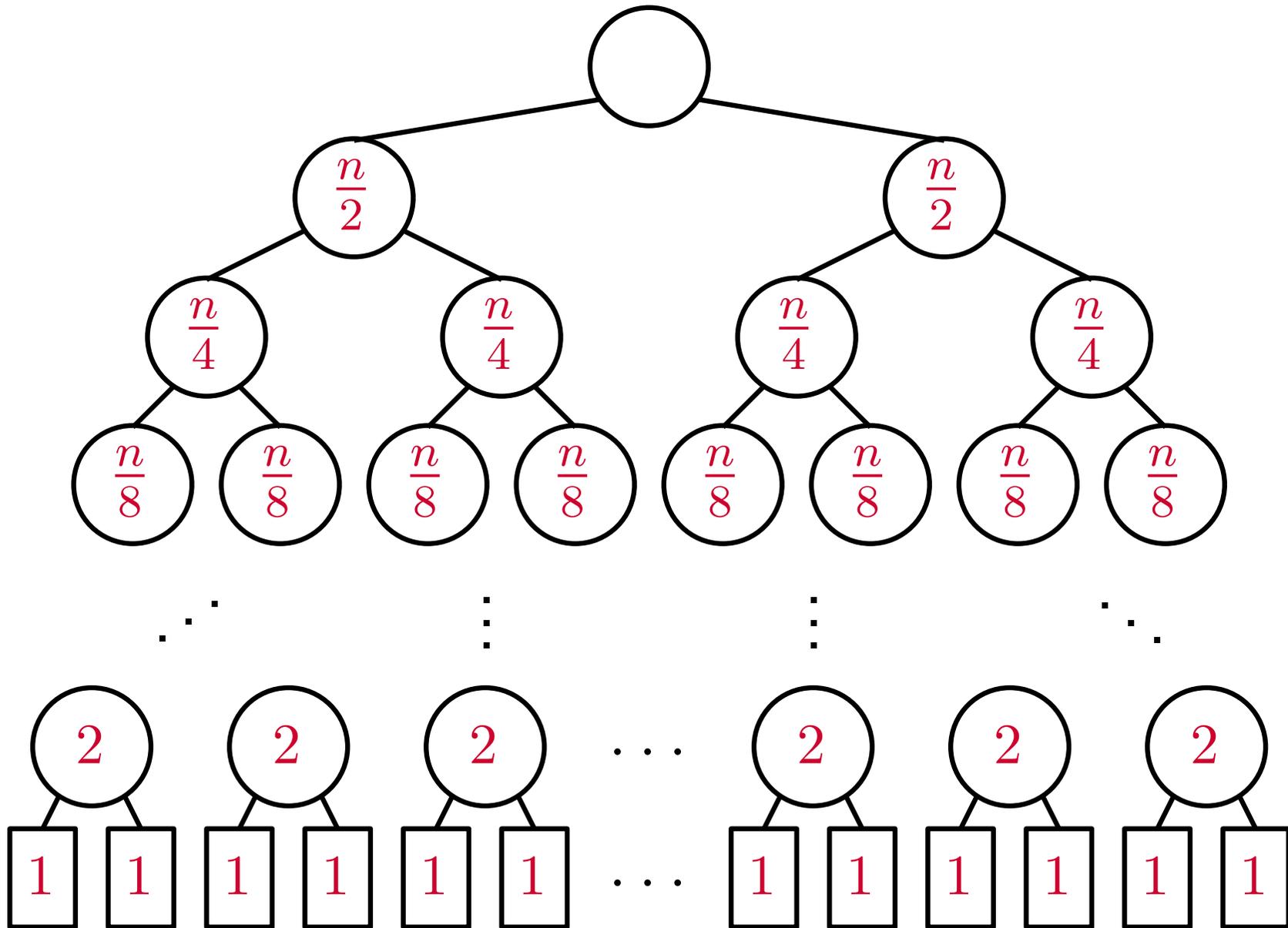
$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$

2

1



Complejidad Merge Sort

Tamaño arreglo

Piezas

¿# de operaciones?

n

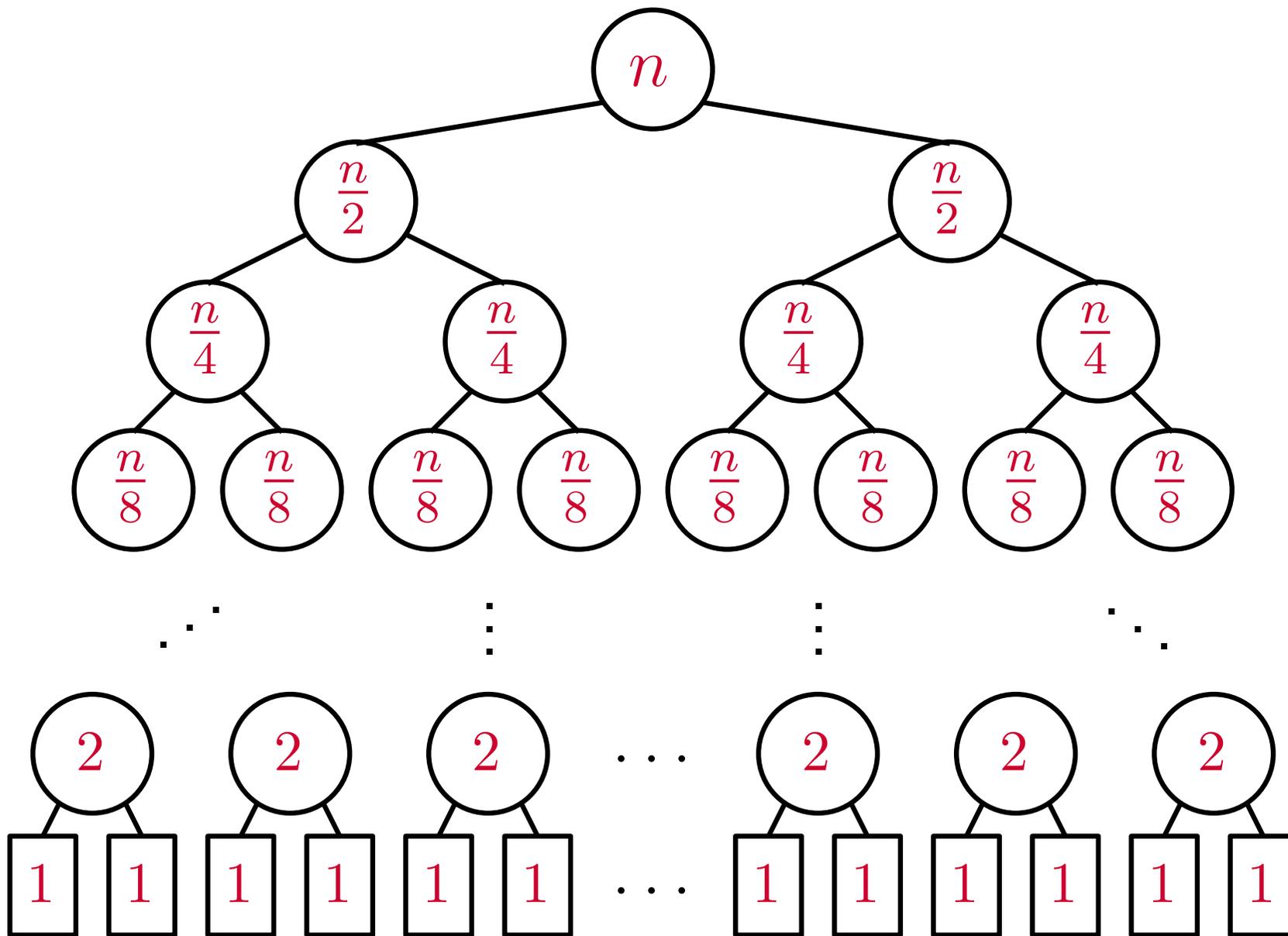
$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$

2

1

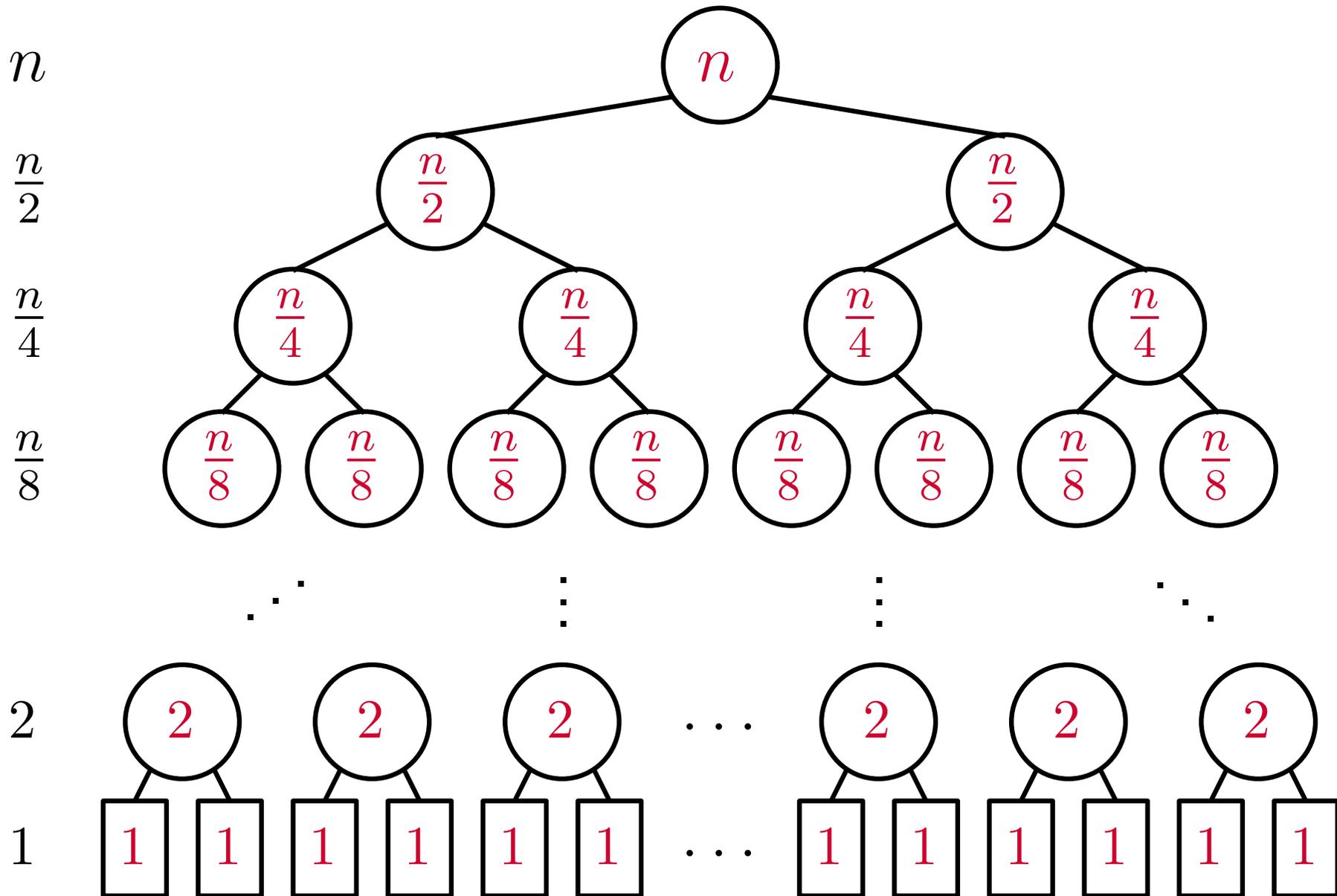


Complejidad Merge Sort

Tamaño arreglo

Sumar lo rojo

Piezas



Complejidad Merge Sort

Tamaño arreglo

Piezas

Sumar lo rojo

$$n + n + \dots + n$$

n

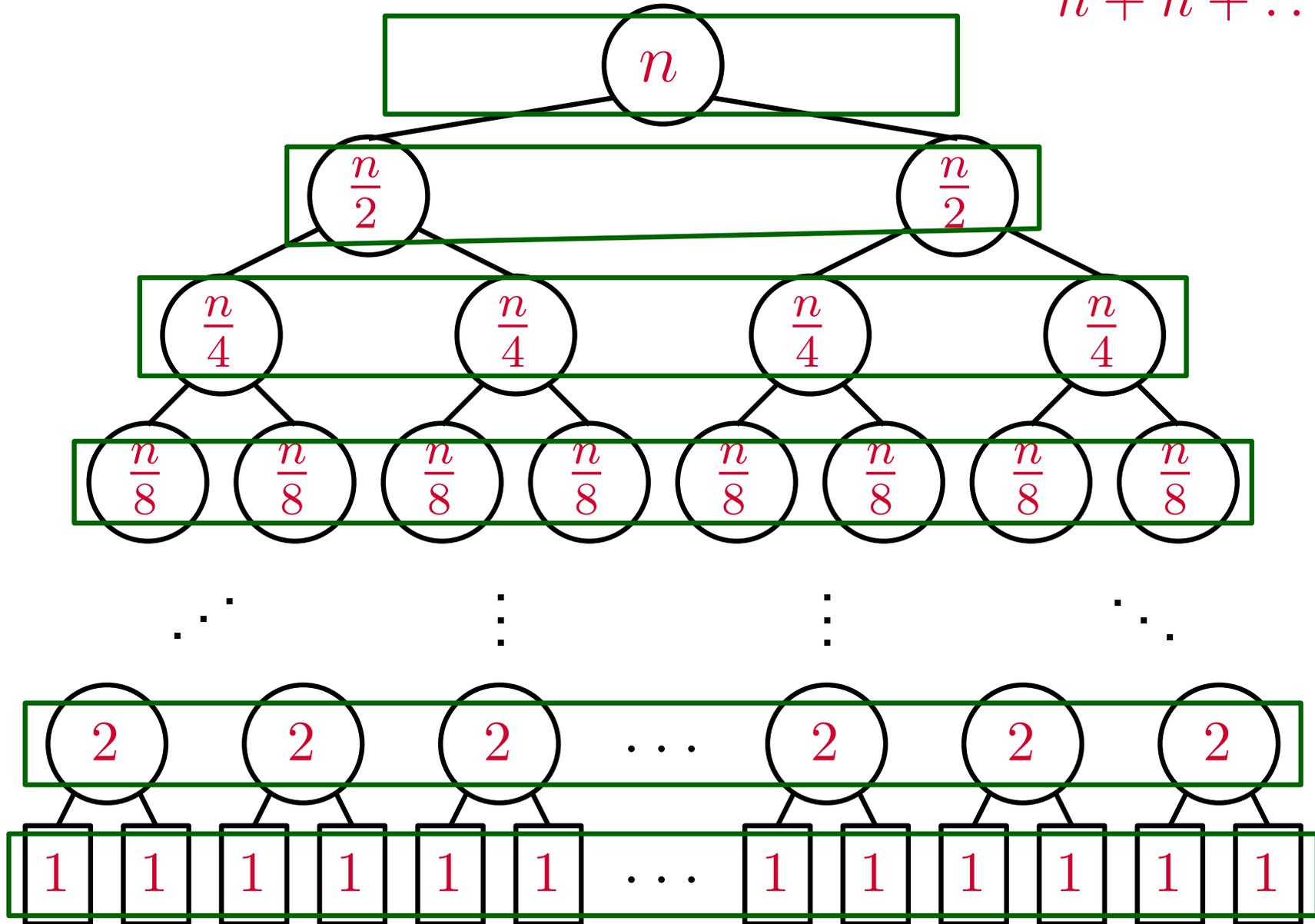
$\frac{n}{2}$

$\frac{n}{4}$

$\frac{n}{8}$

2

1



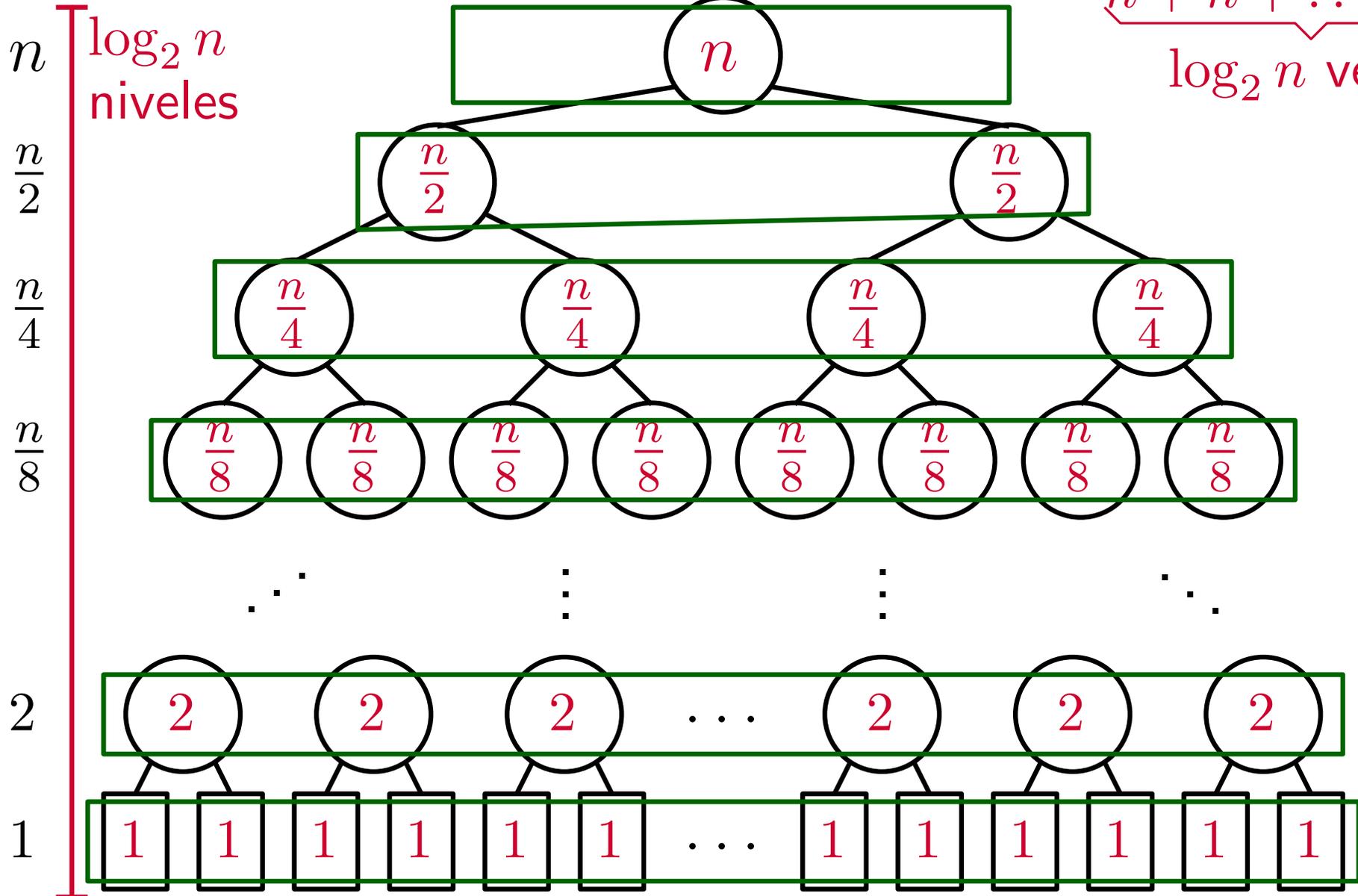
Complejidad Merge Sort

Tamaño arreglo

Piezas

Sumar lo rojo

$$\underbrace{n + n + \dots + n}_{\log_2 n \text{ veces}}$$



Complejidad Merge Sort

Tamaño arreglo

Piezas

Sumar lo rojo

$$\underbrace{n + n + \dots + n}_{\log_2 n \text{ veces}}$$

