



# Алгоритмы поиска и сортировки

ЕМЕЛЬЯНОВ ЭДУАРД ПАВЛОВИЧ

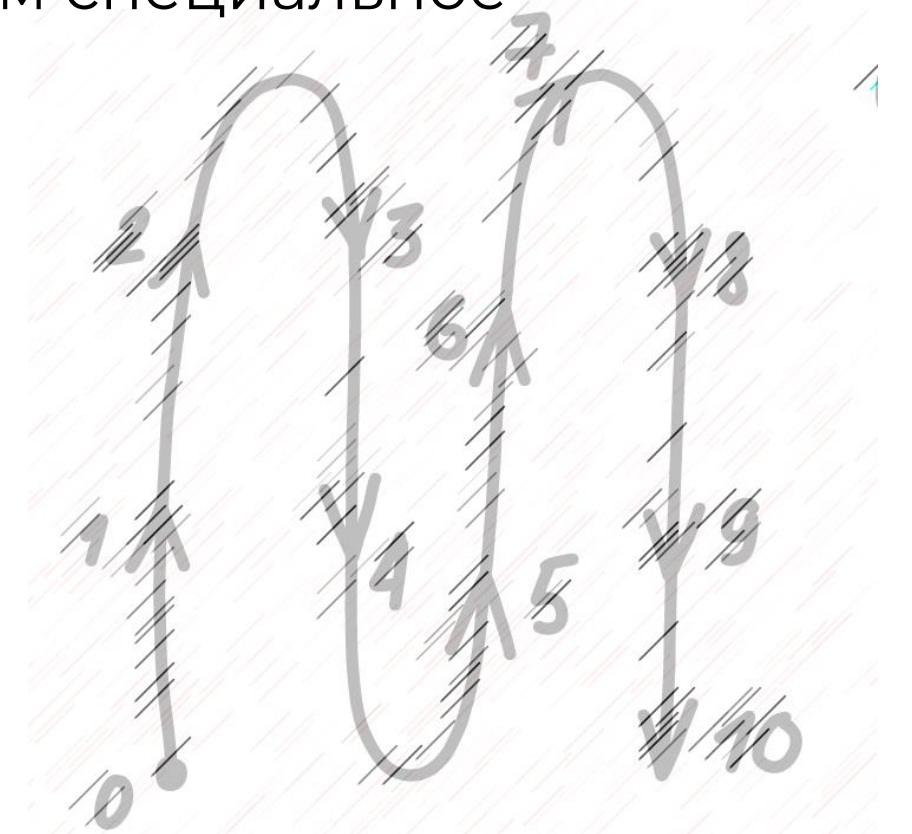




# Поиск в неупорядоченных массивах



- 1) **Начало.** Получаем на вход массив и искомое значение
- 2) **Итерация.** Последовательно перебираем элементы массива, начиная с первого
- 3) **Сравнение.** Для каждого элемента проверяем, равен ли он искомому значению
- 4) **Результат.**
  - Если найден совпадающий элемент – возвращаем его позицию (индекс)
  - Если весь массив просмотрен и элемент не найден – возвращаем специальное значение (обычно -1 или *None*)



# Поиск в неупорядоченных массивах



7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

5

7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

↑  
 $7 \neq 5 \rightarrow$  дальше

7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

↑  
 $2 \neq 5 \rightarrow$  дальше

7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

↑  
 $9 \neq 5 \rightarrow$  дальше

# Поиск в неупорядоченных массивах



7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

5

7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

↑  
 $4 \neq 5 \rightarrow$  дальше

7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

↑  
 $5 = 5 \rightarrow$  КОНЕЦ

# Поиск в неупорядоченных массивах



7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

5

7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

↑  
 $4 \neq 5 \rightarrow$  дальше

Худший случай:  $O(n)$

7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

↑  
 $5 = 5 \rightarrow$  КОНЕЦ

# Поиск в неупорядоченных массивах



7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

5

7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

↑  
 $4 \neq 5 \rightarrow$  дальше

7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

↑  
 $5 = 5 \rightarrow$  КОНЕЦ

Худший случай:  $O(n)$

Средний случай:  $O(n/2) \approx O(n)$

# Поиск в неупорядоченных массивах



7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

5

7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

↑  
 $4 \neq 5 \rightarrow$  дальше

7	2	9	4	5	8	3	6
---	---	---	---	---	---	---	---

↑  
 $5 = 5 \rightarrow$  КОНЕЦ

Худший случай:  $O(n)$

Средний случай:  $O(n/2) \approx O(n)$

Лучший случай:  $O(1)$



# Дихотомический поиск

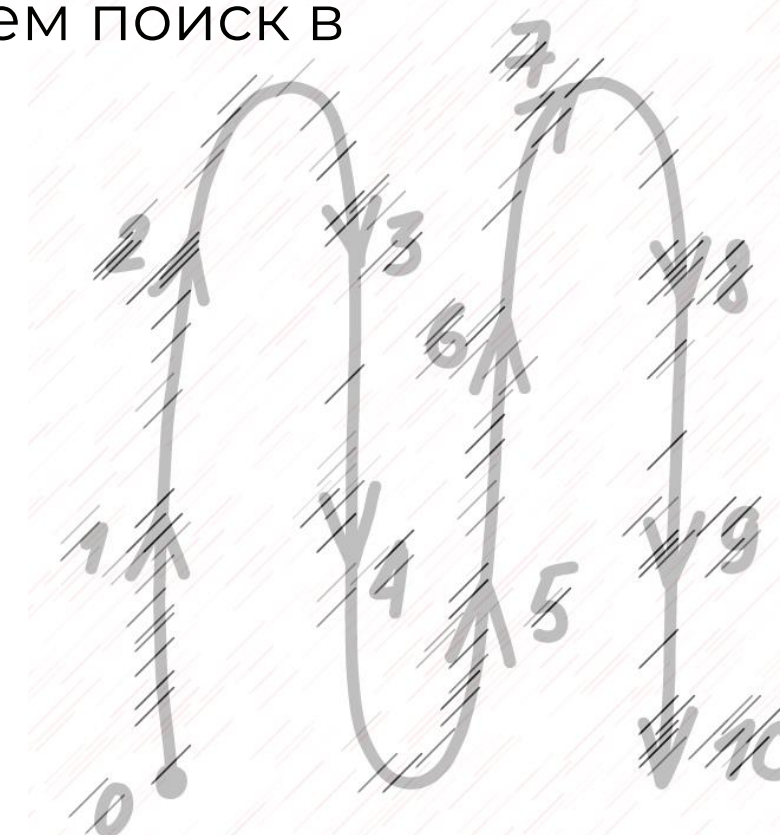


1) Находим средний элемент массива

2) Сравниваем его с искомым значением

3) **Результат.**

- Если средний элемент равен искомому – поиск завершен
- Если искомое значение **меньше** среднего элемента – продолжаем поиск в левой половине (п.1)
- Если искомое значение **больше** среднего элемента – продолжаем поиск в правой половине (п.1)





# Дихотомический поиск



1	3	5	7	9	11	13	15	17	19
---	---	---	---	---	----	----	----	----	----

13
----

1	3	5	7	9	11	13	15	17	19
0	1	2	3	4	5	6	7	8	9

$middle = (left + right) // 2 = (0 + 9) // 2 = 4$

$array[4] = 9 \rightarrow 9 < 13$  элемент СПРАВА от середины

# Дихотомический поиск



1	3	5	7	9	11	13	15	17	19
---	---	---	---	---	----	----	----	----	----

13

1	3	5	7	9	11	13	15	17	19
0	1	2	3	4	5	6	7	8	9

$$\text{middle} = (5 + 9) // 2 = 7$$

$\text{array}[7] = 15 \rightarrow 15 > 13$  элемент СЛЕВА от середины

# Дихотомический поиск



1	3	5	7	9	11	13	15	17	19
---	---	---	---	---	----	----	----	----	----

13

1	3	5	7	9	11	13	15	17	19
0	1	2	3	4	5	6	7	8	9

$$\text{middle} = (5 + 6) // 2 = 5$$

$\text{array}[5] = 11 \rightarrow 11 < 13$  элемент СПРАВА от середины

# Дихотомический поиск



1	3	5	7	9	11	13	15	17	19
---	---	---	---	---	----	----	----	----	----

13

1	3	5	7	9	11	13	15	17	19
0	1	2	3	4	5	6	7	8	9

$middle = (6 + 6) // 2 = 6$

$array[6] = 13 \rightarrow \text{КОНЕЦ}$



# Дихотомический поиск



1	3	5	7	9	11	13	15	17	19
---	---	---	---	---	----	----	----	----	----

13

1	3	5	7	9	11	13	15	17	19
0	1	2	3	4	5	6	7	8	9

$middle = (6 + 6) // 2 = 6$

$array[6] = 13 \rightarrow \text{КОНЕЦ}$

Худший случай:  $O(\log n)$

# Дихотомический поиск



1	3	5	7	9	11	13	15	17	19
---	---	---	---	---	----	----	----	----	----

13

1	3	5	7	9	11	13	15	17	19
0	1	2	3	4	5	6	7	8	9

$middle = (6 + 6) // 2 = 6$

$array[6] = 13 \rightarrow \text{КОНЕЦ}$

Худший случай:  $O(\log n)$

Средний случай:  $O(\log n)$

# Дихотомический поиск



1	3	5	7	9	11	13	15	17	19
---	---	---	---	---	----	----	----	----	----

13

1	3	5	7	9	11	13	15	17	19
0	1	2	3	4	5	6	7	8	9

$middle = (6 + 6) // 2 = 6$

$array[6] = 13 \rightarrow \text{КОНЕЦ}$

Худший случай:  $O(\log n)$

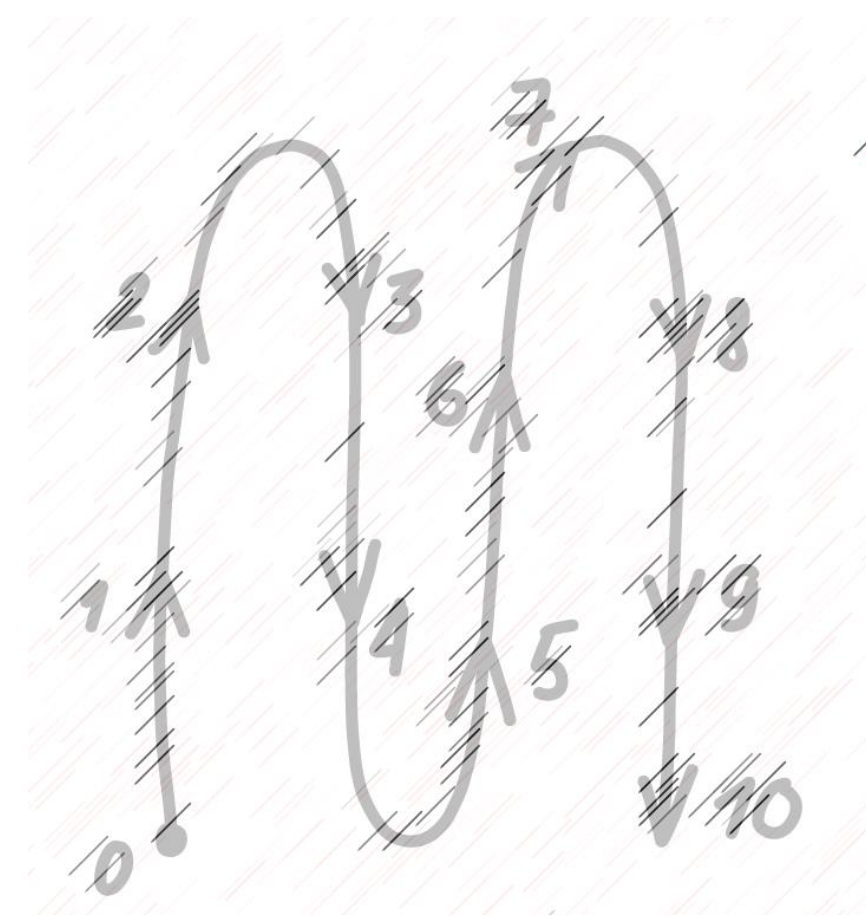
Средний случай:  $O(\log n)$

Лучший случай:  $O(1)$

# Пузырьковая сортировка (Bubble Sort)



- 1) Проходим по массиву от начала до конца
- 2) Сравниваем каждую пару соседних элементов
- 3) Если элементы расположены в неправильном порядке (предыдущий больше следующего), меняем их местами
- 4) Повторяем процесс до тех пор, пока массив не будет полностью отсортирован





# Пузырьковая сортировка (Bubble Sort)



64	34	25	12	22	11	90
----	----	----	----	----	----	----

Итерация I

64	34	25	12	22	11	90
----	----	----	----	----	----	----

34	64	25	12	22	11	90
----	----	----	----	----	----	----

34	25	12	22	64	11	90
----	----	----	----	----	----	----

34	25	64	12	22	11	90
----	----	----	----	----	----	----

34	25	12	22	11	64	90
----	----	----	----	----	----	----

34	25	12	64	22	11	90
----	----	----	----	----	----	----

34	25	12	22	11	64	90
----	----	----	----	----	----	----

# Пузырьковая сортировка (Bubble Sort)



64	34	25	12	22	11	90
----	----	----	----	----	----	----

Итерация II

34	25	12	22	11	64	90
----	----	----	----	----	----	----

25	34	12	22	11	64	90
----	----	----	----	----	----	----

25	12	22	11	34	64	90
----	----	----	----	----	----	----

25	12	34	22	11	64	90
----	----	----	----	----	----	----

25	12	22	11	34	64	90
----	----	----	----	----	----	----

25	12	22	34	11	64	90
----	----	----	----	----	----	----

25	12	22	11	34	64	90
----	----	----	----	----	----	----

# Пузырьковая сортировка (Bubble Sort)



64	34	25	12	22	11	90
----	----	----	----	----	----	----

25	12	22	11	34	64	90
----	----	----	----	----	----	----

Итерация III

12	22	11	25	34	64	90
----	----	----	----	----	----	----

Итерация IV

12	11	22	25	34	64	90
----	----	----	----	----	----	----

Итерация IV

11	12	22	25	34	64	90
----	----	----	----	----	----	----

Итерация V

# Пузырьковая сортировка (Bubble Sort)



64	34	25	12	22	11	90
----	----	----	----	----	----	----

Худший случай:  $O(n^2)$

25	12	22	11	34	64	90
----	----	----	----	----	----	----

Итерация III

12	22	11	25	34	64	90
----	----	----	----	----	----	----

Итерация IV

12	11	22	25	34	64	90
----	----	----	----	----	----	----

Итерация IV

11	12	22	25	34	64	90
----	----	----	----	----	----	----

Итерация V



# Пузырьковая сортировка (Bubble Sort)



64	34	25	12	22	11	90
----	----	----	----	----	----	----

Худший случай:  $O(n^2)$

Средний случай:  $O(n^2)$

25	12	22	11	34	64	90
----	----	----	----	----	----	----

Итерация III

12	22	11	25	34	64	90
----	----	----	----	----	----	----

Итерация IV

12	11	22	25	34	64	90
----	----	----	----	----	----	----

Итерация IV

11	12	22	25	34	64	90
----	----	----	----	----	----	----

Итерация V

# Пузырьковая сортировка (Bubble Sort)



64	34	25	12	22	11	90
----	----	----	----	----	----	----

25	12	22	11	34	64	90
----	----	----	----	----	----	----

Итерация III

12	22	11	25	34	64	90
----	----	----	----	----	----	----

Итерация IV

12	11	22	25	34	64	90
----	----	----	----	----	----	----

Итерация IV

11	12	22	25	34	64	90
----	----	----	----	----	----	----

Итерация V

Худший случай:  $O(n^2)$

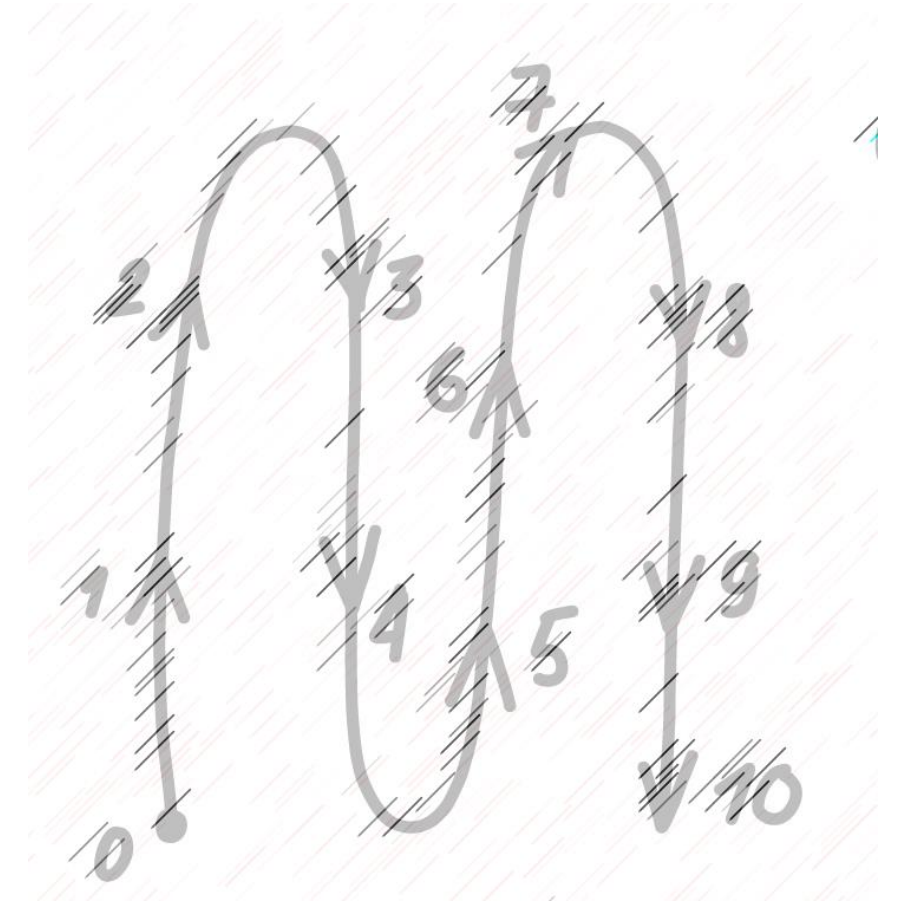
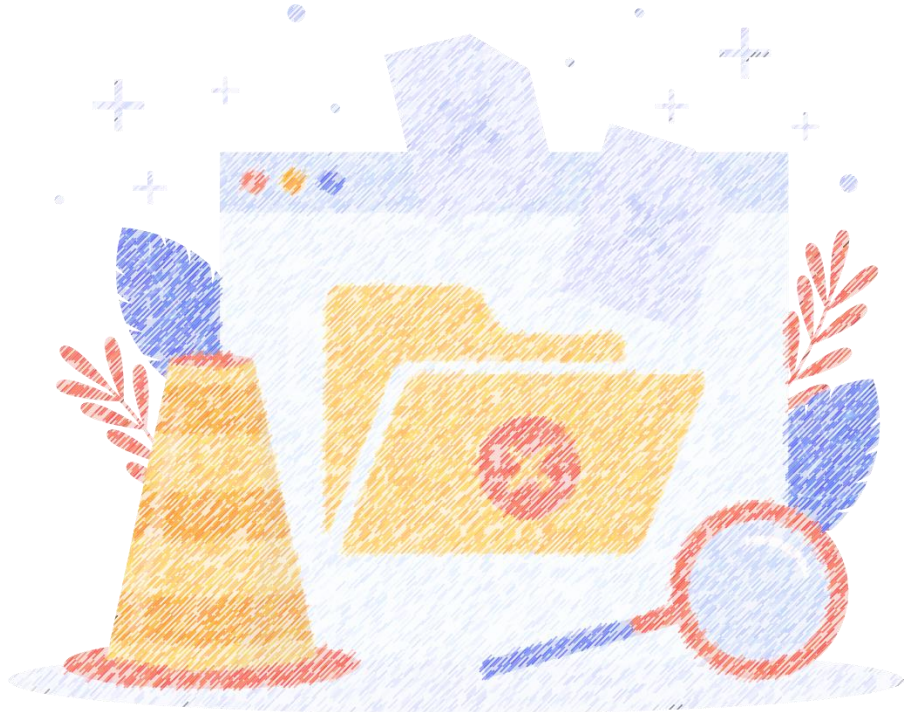
Средний случай:  $O(n^2)$

Лучший случай:  $O(n)$

# Сортировка слиянием (Merge Sort)



- 1) **Разделение (Divide).** Рекурсивно делим массив на две половины, пока не получим подмассивы размером 1 элемент (которые по определению являются отсортированными)
- 2) **Властвование (Conquer).** Рекурсивно сортируем подмассивы
- 3) **Слияние (Combine).** Объединяем отсортированные подмассивы в один отсортированный массив



# Сортировка слиянием (Merge Sort)



38	27	43	3	9	82	10
----	----	----	---	---	----	----

38	27	43	3	9	82	10
----	----	----	---	---	----	----

38	27	43	3
----	----	----	---

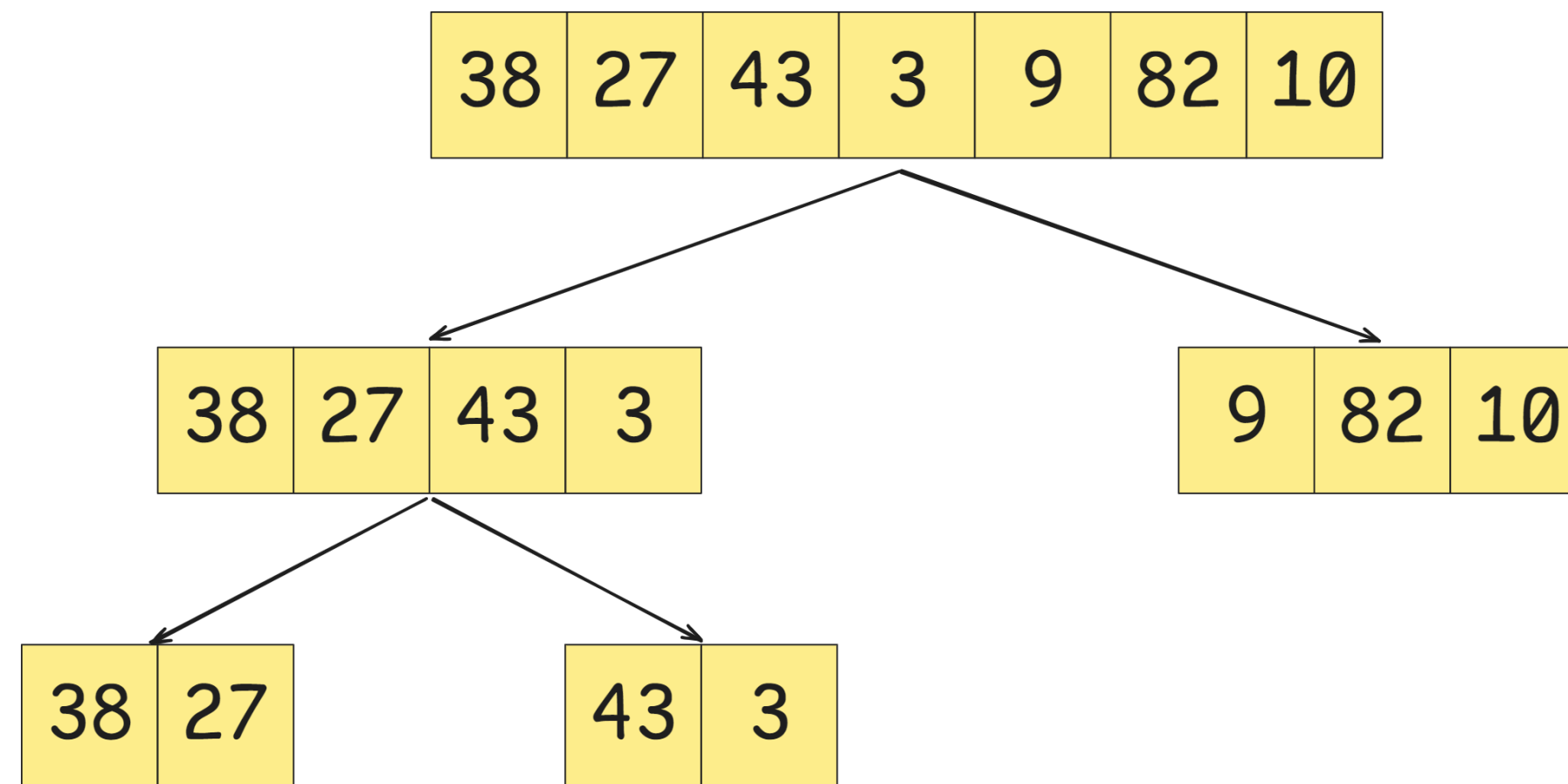
9	82	10
---	----	----



# Сортировка слиянием (Merge Sort)



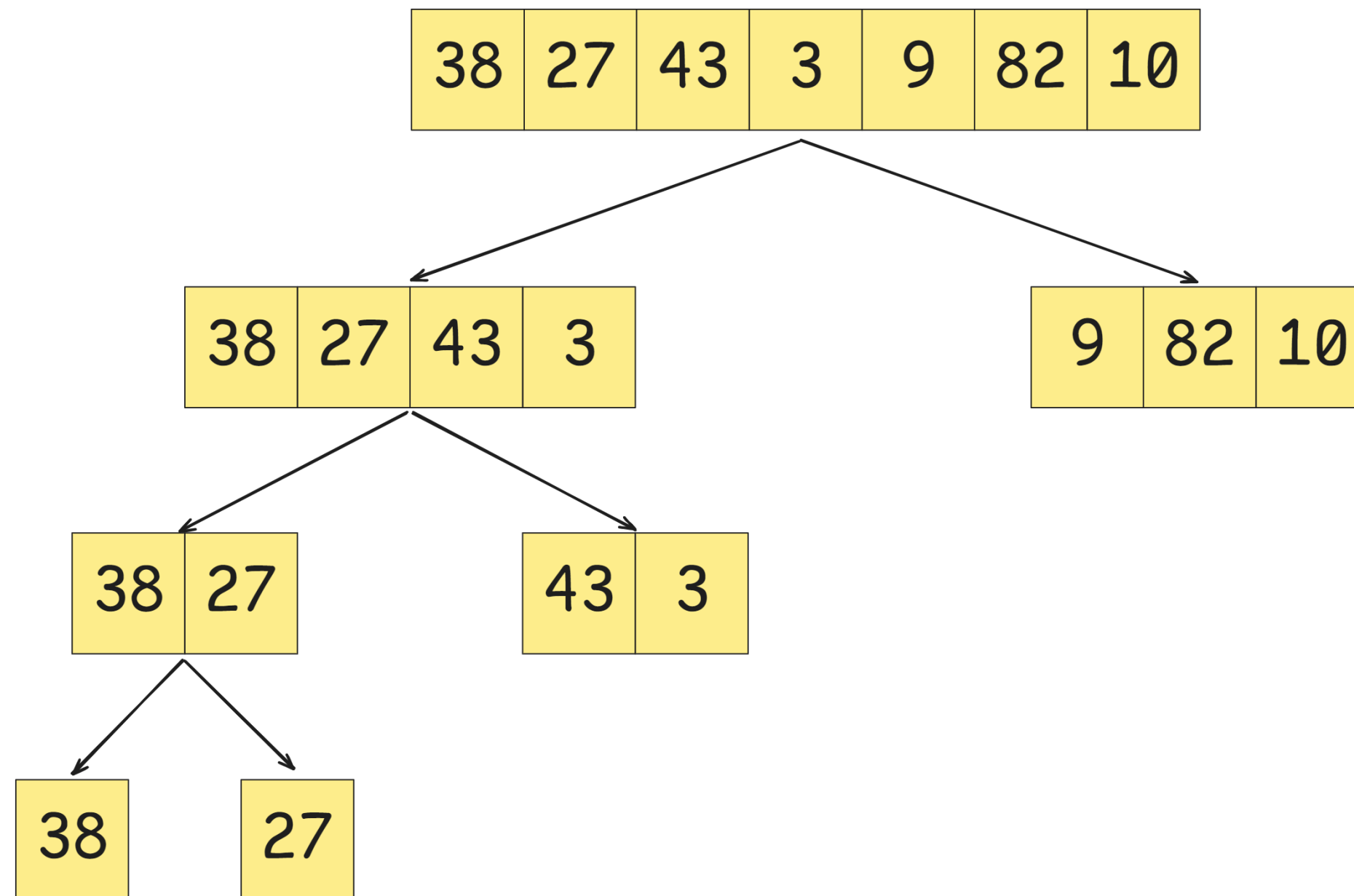
38	27	43	3	9	82	10
----	----	----	---	---	----	----



# Сортировка слиянием (Merge Sort)



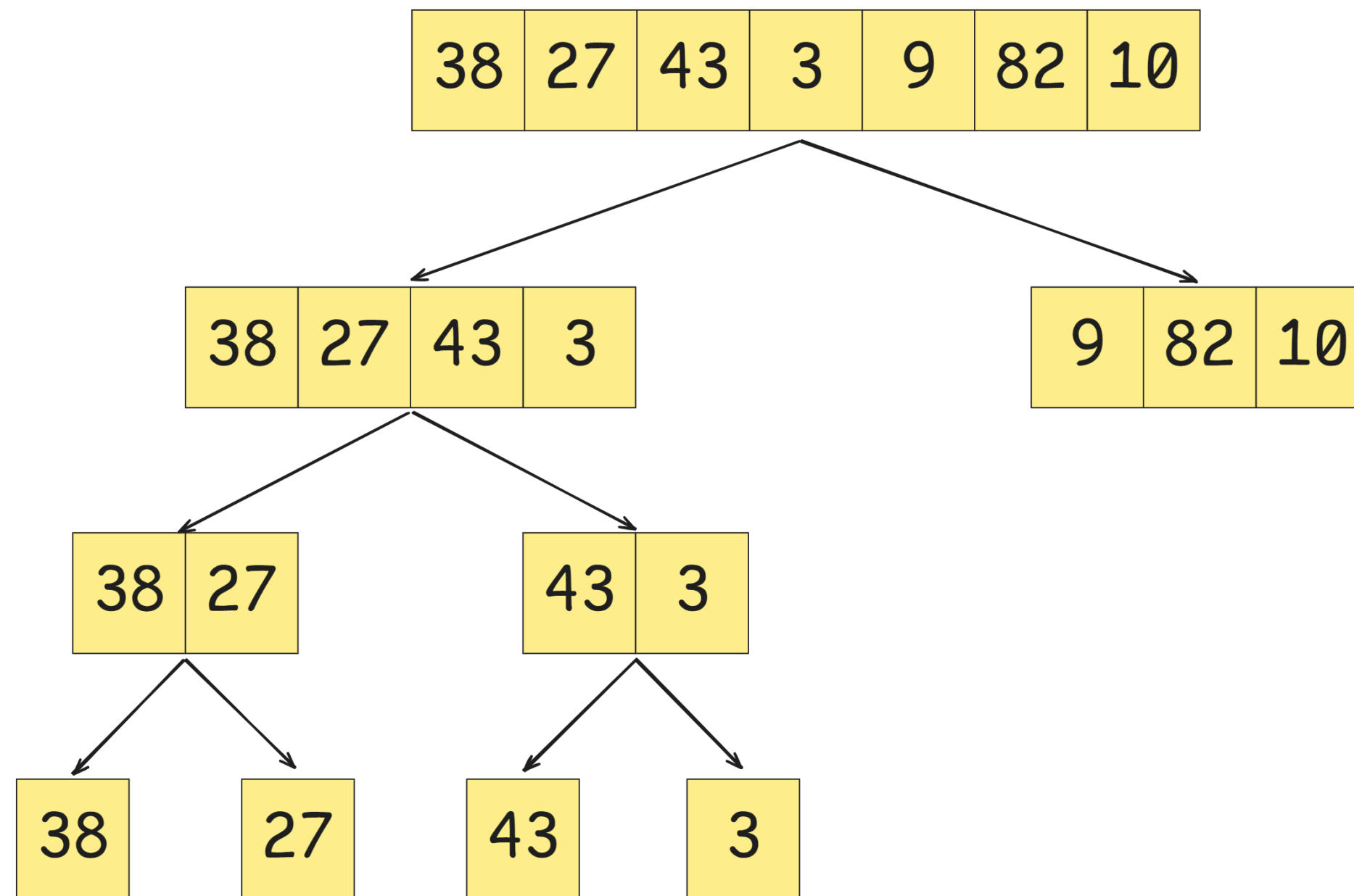
38	27	43	3	9	82	10
----	----	----	---	---	----	----



# Сортировка слиянием (Merge Sort)



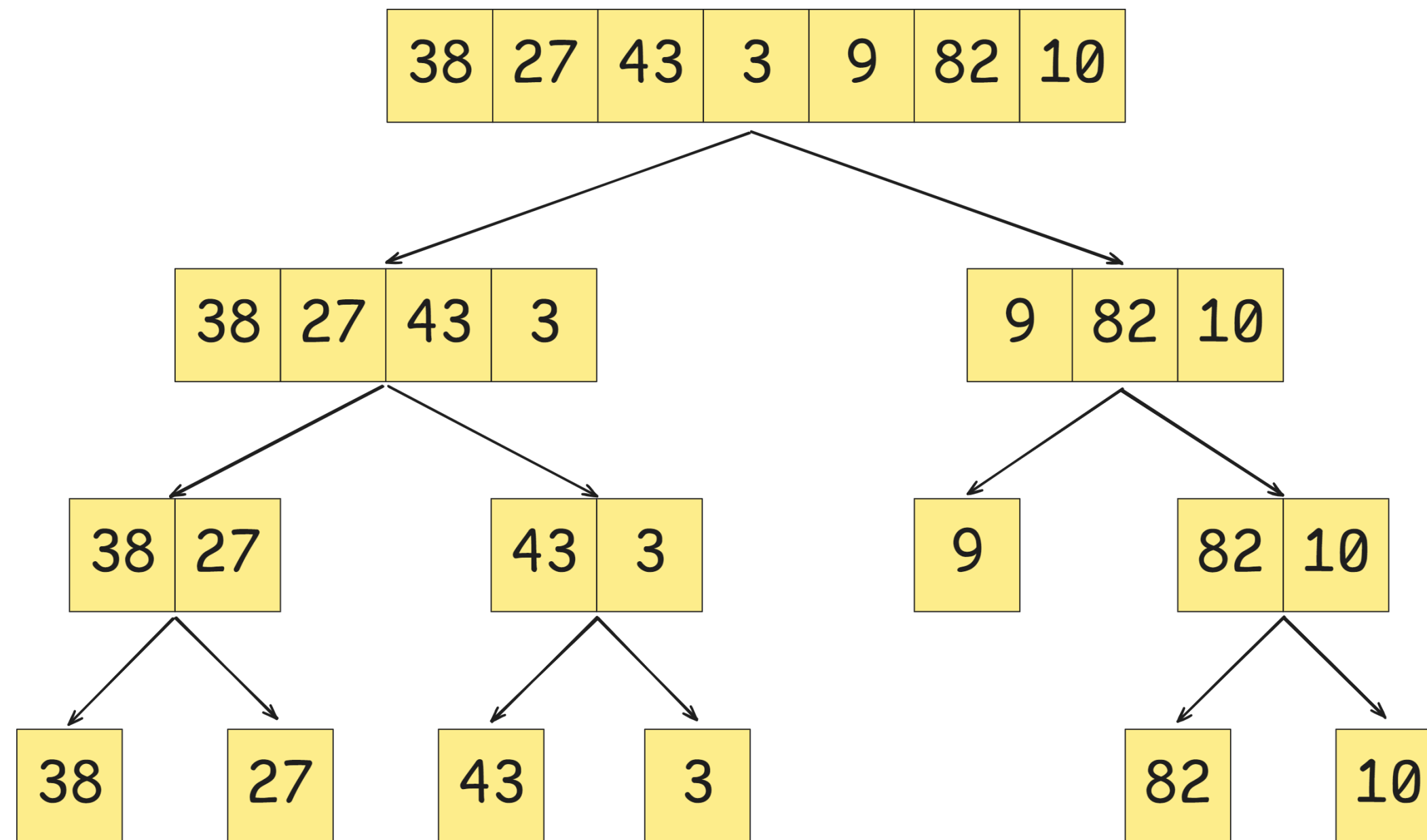
38	27	43	3	9	82	10
----	----	----	---	---	----	----



# Сортировка слиянием (Merge Sort)



38	27	43	3	9	82	10
----	----	----	---	---	----	----

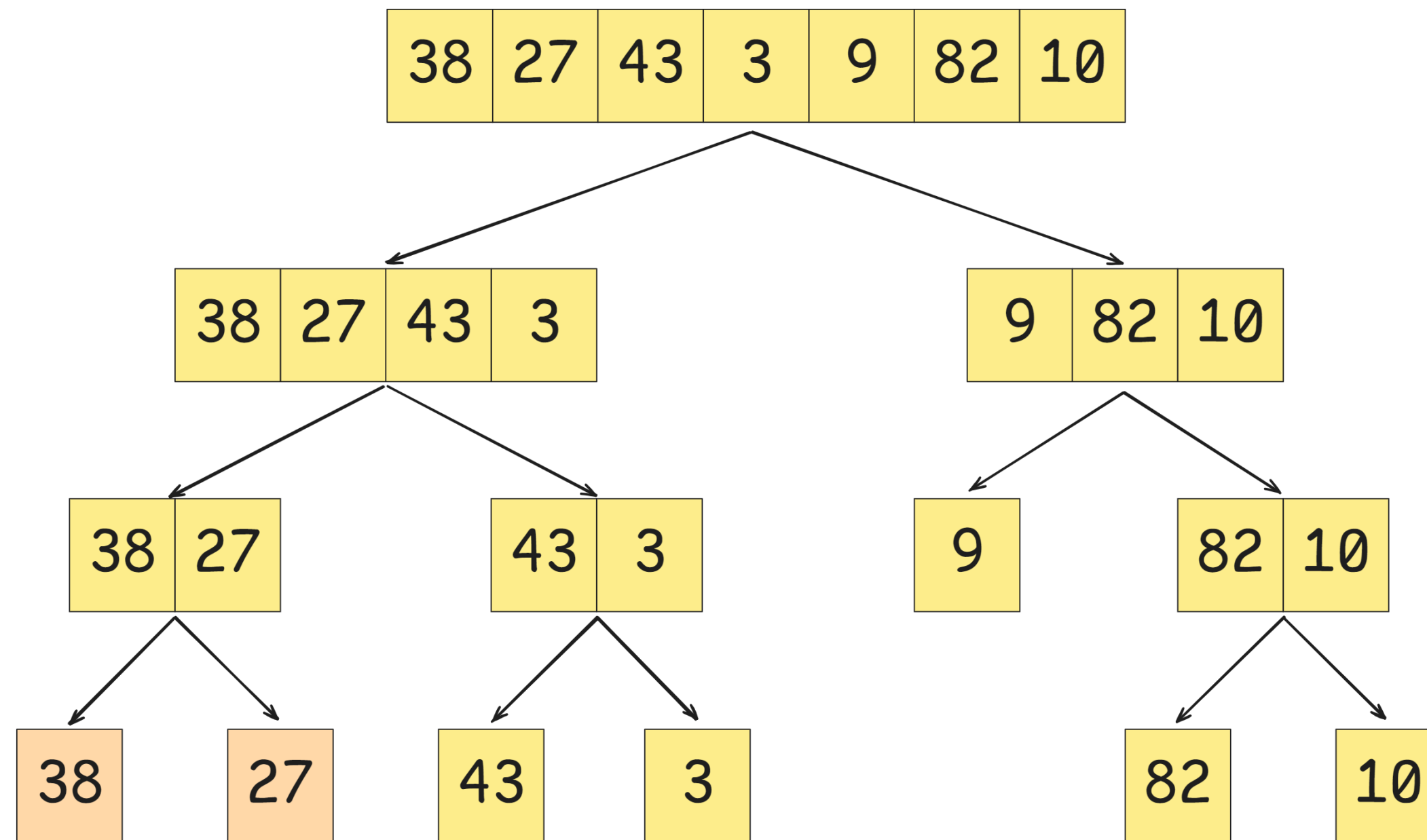


# Сортировка слиянием (Merge Sort)



38	27	43	3	9	82	10
----	----	----	---	---	----	----

Слияние



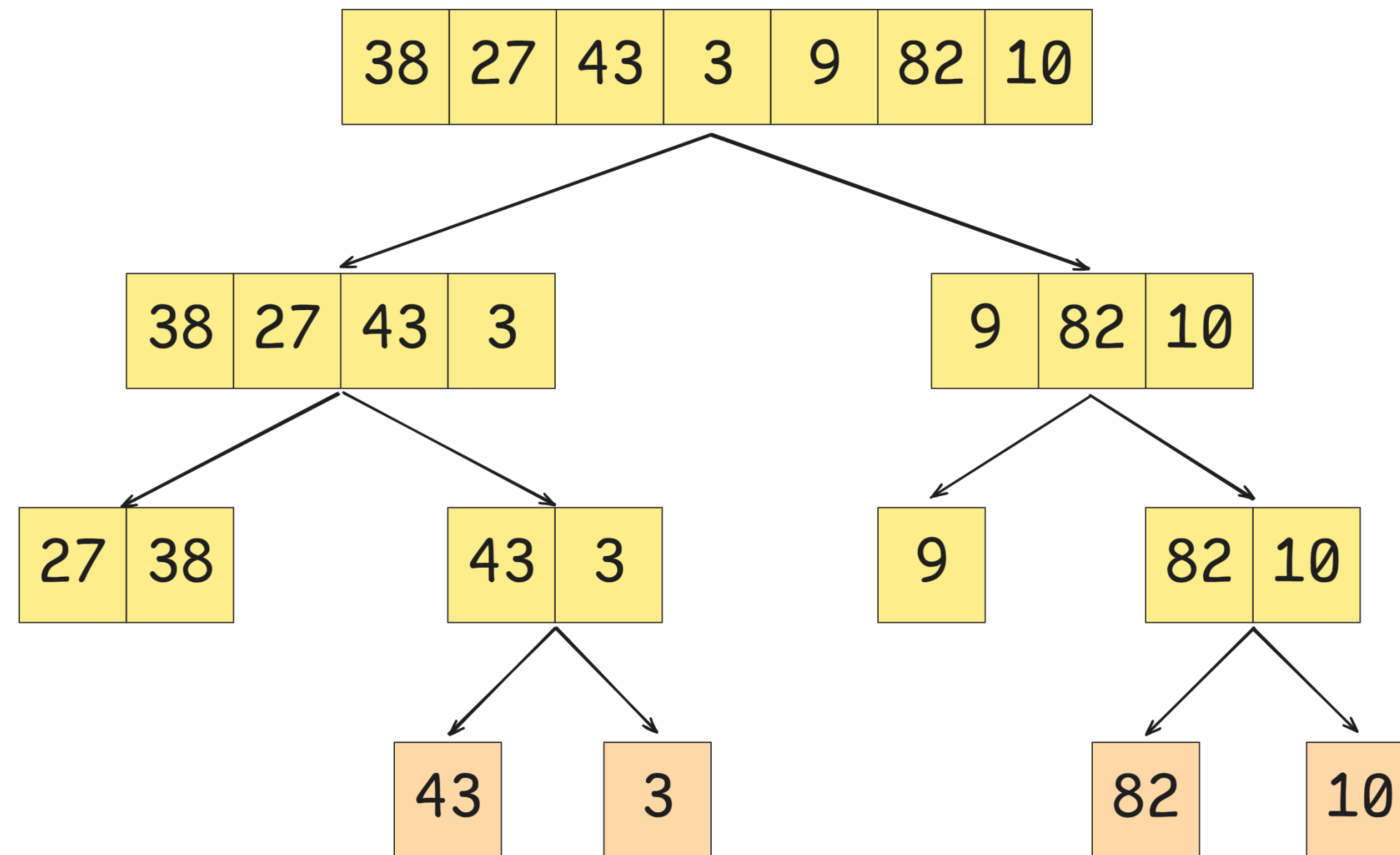


# Сортировка слиянием (Merge Sort)



38	27	43	3	9	82	10
----	----	----	---	---	----	----

Слияние

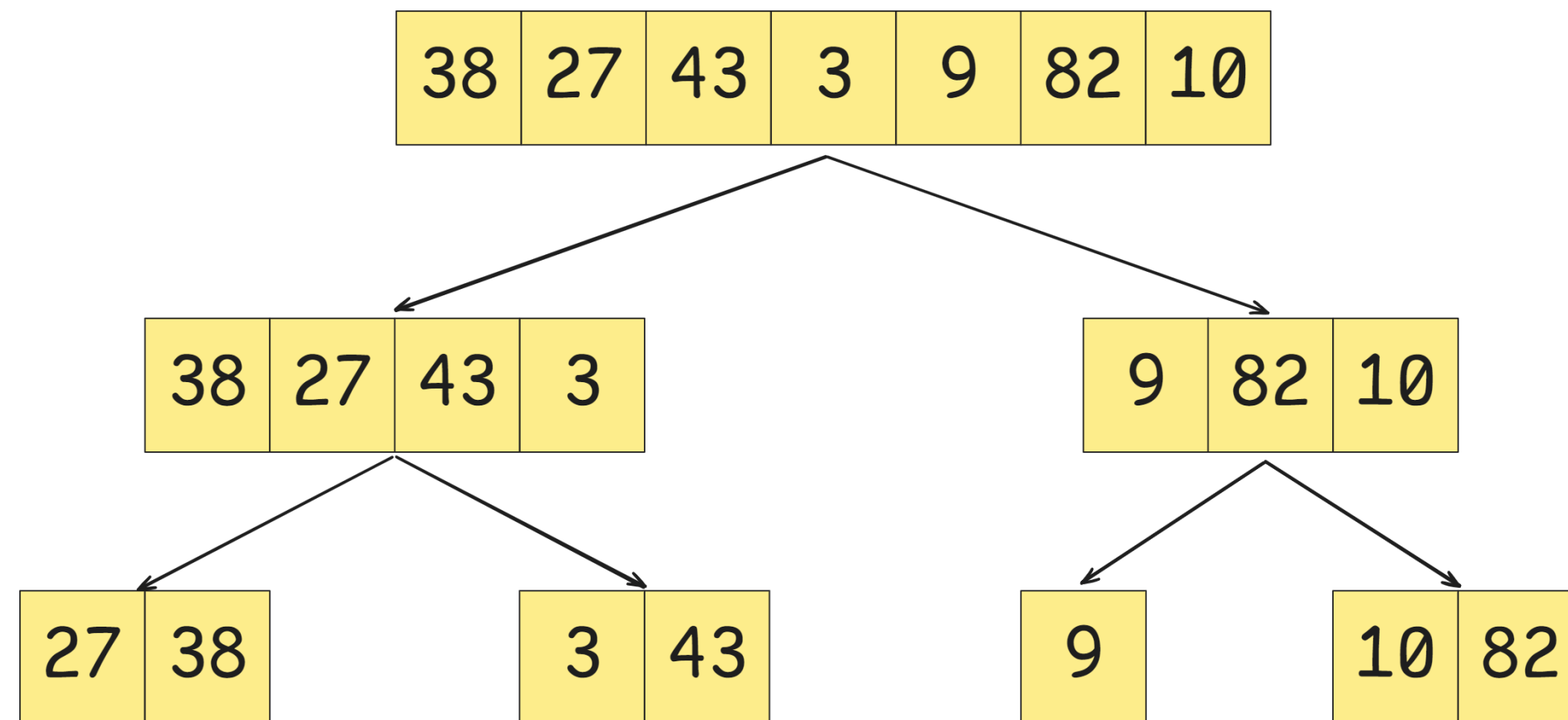


# Сортировка слиянием (Merge Sort)



38	27	43	3	9	82	10
----	----	----	---	---	----	----

Слияние

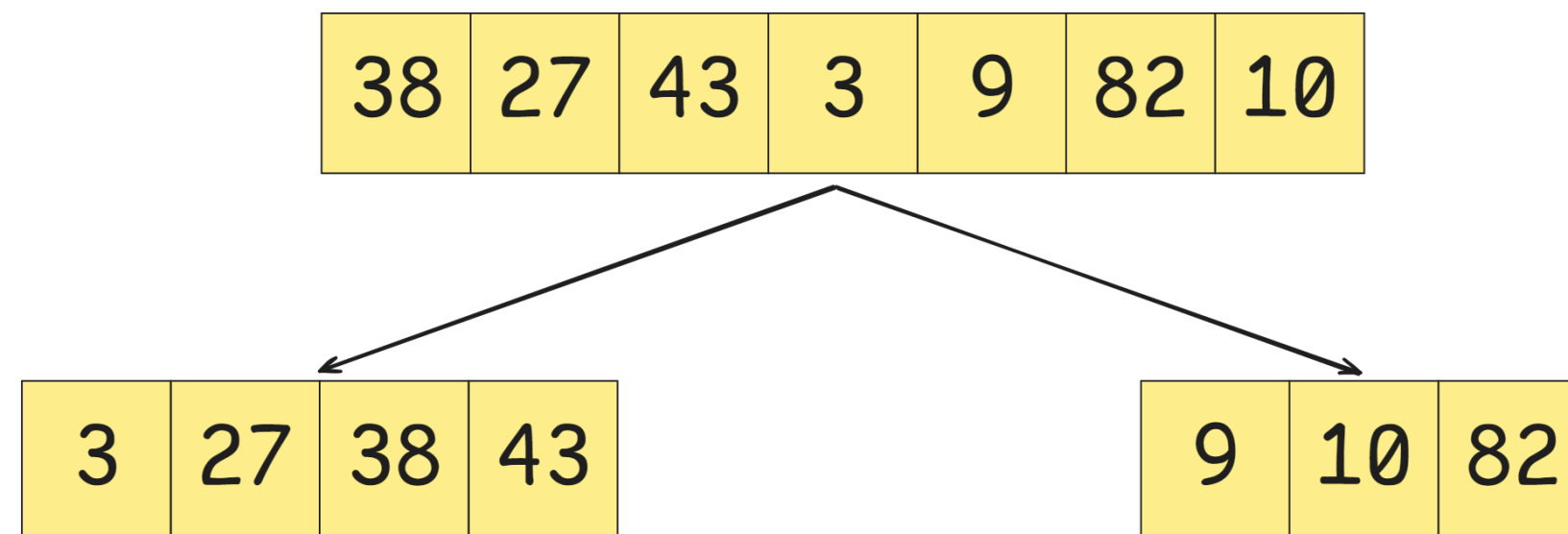


# Сортировка слиянием (Merge Sort)



38	27	43	3	9	82	10
----	----	----	---	---	----	----

Слияние



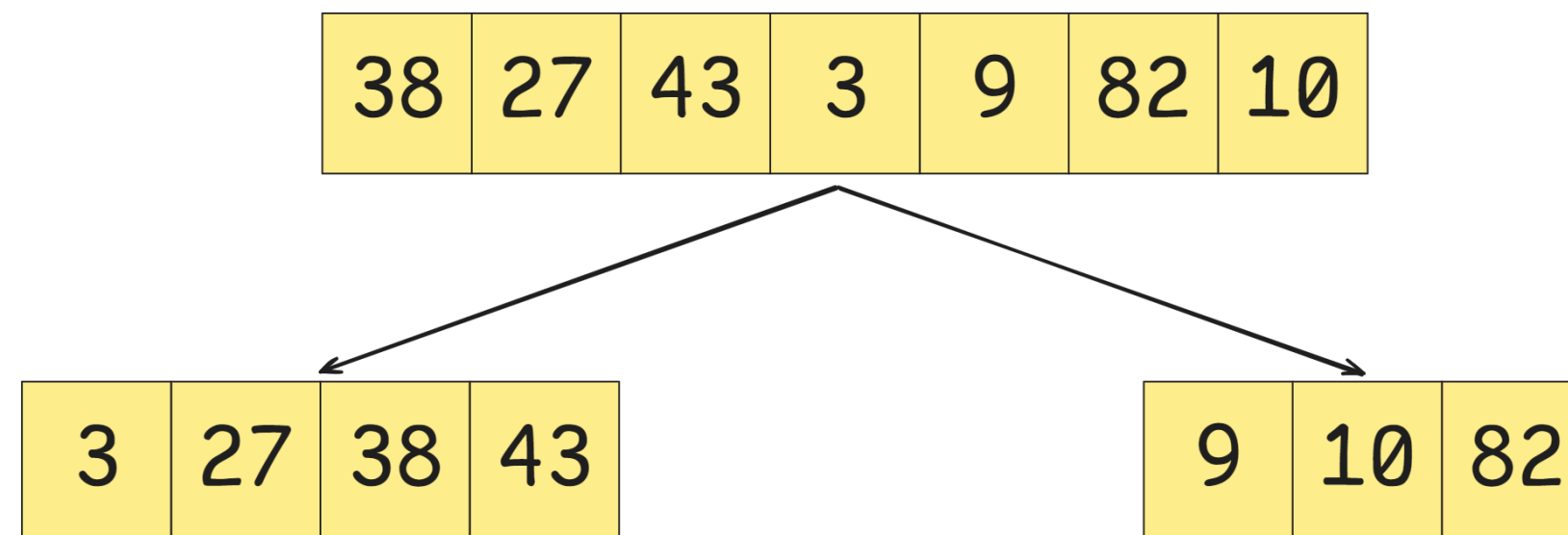
3	9	10	27	38	43	82
---	---	----	----	----	----	----

# Сортировка слиянием (Merge Sort)



38	27	43	3	9	82	10
----	----	----	---	---	----	----

Слияние



Худший случай:  $O(n * \log n)$

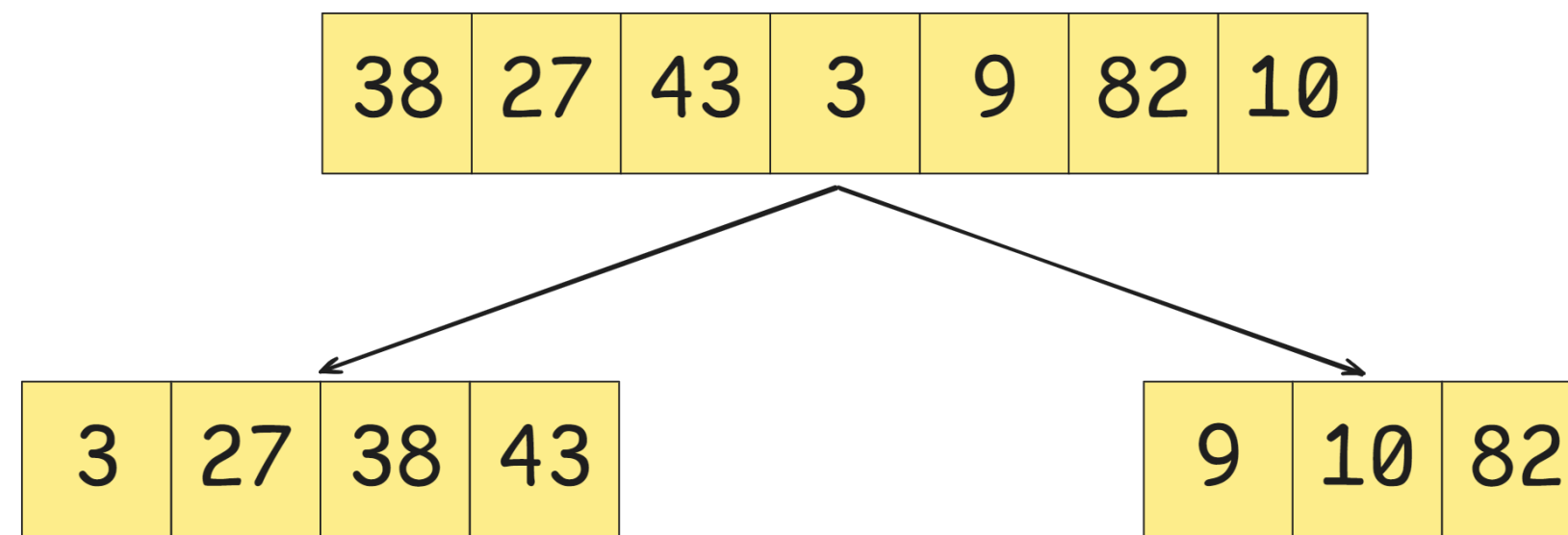
3	9	10	27	38	43	82
---	---	----	----	----	----	----

# Сортировка слиянием (Merge Sort)



38	27	43	3	9	82	10
----	----	----	---	---	----	----

Слияние



Худший случай:  $O(n * \log n)$

Средний случай:  $O(n * \log n)$

3	9	10	27	38	43	82
---	---	----	----	----	----	----

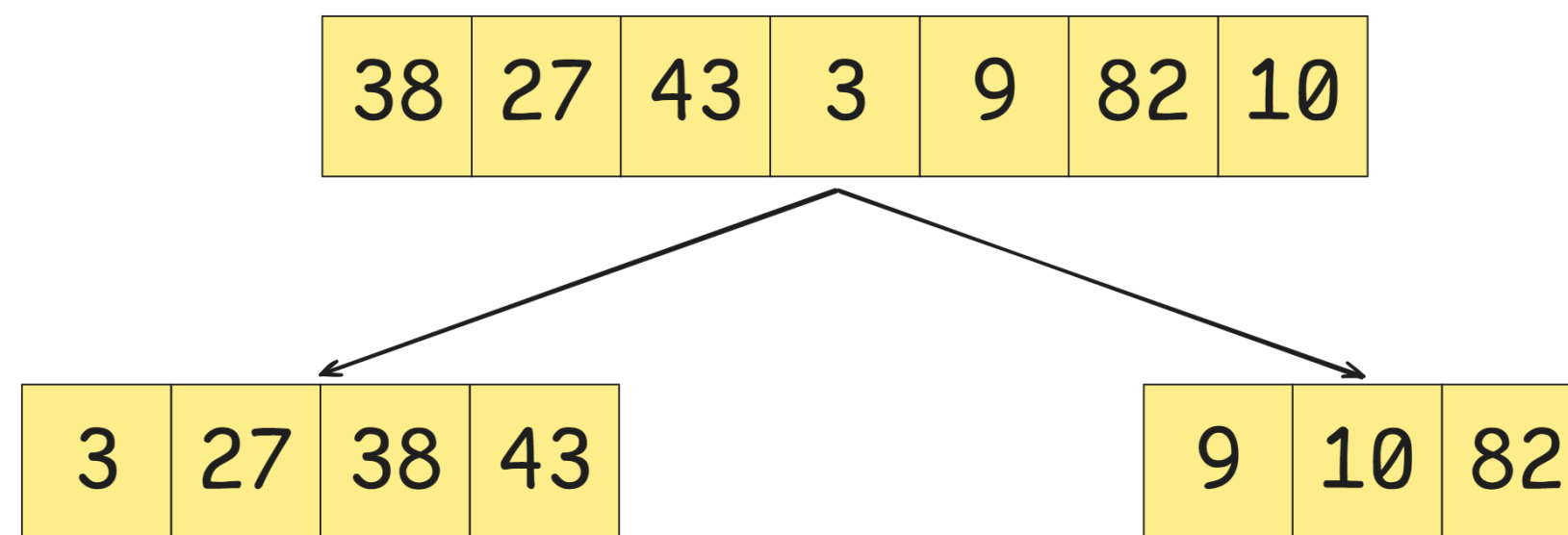


# Сортировка слиянием (Merge Sort)



38	27	43	3	9	82	10
----	----	----	---	---	----	----

Слияние



Худший случай:  $O(n * \log n)$

Средний случай:  $O(n * \log n)$

Лучший случай:  $O(n * \log n)$

3	9	10	27	38	43	82
---	---	----	----	----	----	----

# Алгоритм Кнута-Морриса-Пратта (КМП)



a	b	c	a	b	a	b	d	a	b	a	a	b	a
a	b	a	a	b	a								
		a	b	a	a	b	a						
			a	b	a	a	b	a					
					a	b	a	a	b	a			
							a	b	a	a	b	a	
								a	b	a	a	b	a



**Спасибо за внимание**