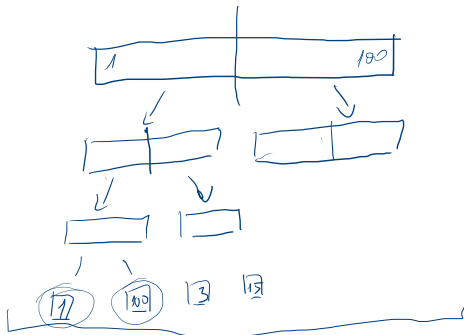


Сортировки методом декомпозиции

Метод декомпозиции = "разделяй и властвуй"

Суть: сложная задача разбивается на несколько простых, которые подобры исходной, но имеют меньший объем. Далее подзадачи решаются рекурсивно и полученные решения комбинируются для получения ответа.



Сортировка слиянием

Merge sort

Сортировка слиянием

72	4	54	10	41	27	54	3	0
----	---	----	----	----	----	----	---	---

Сортировка слиянием

72	4	54	10	41	27	54	3	0
----	---	----	----	----	----	----	---	---

72	4	54	10
----	---	----	----

41	27	54	3	0
----	----	----	---	---

Сортировка слиянием

72	4	54	10	41	27	54	3	0
----	---	----	----	----	----	----	---	---

72	4	54	10
----	---	----	----

41	27	54	3	0
----	----	----	---	---

72	4
----	---

54	10
----	----

41	27
----	----

54	3	0
----	---	---

Сортировка слиянием

72	4	54	10	41	27	54	3	0
----	---	----	----	----	----	----	---	---

72	4	54	10
----	---	----	----

41	27	54	3	0
----	----	----	---	---

1

72	4
----	---

54	10
----	----

41	27
----	----

54	3	0
----	---	---

2

72

4

54

10

41

27

54

3

0

3

Сортировка слиянием

72	4	54	10	41	27	54	3	0
----	---	----	----	----	----	----	---	---

72	4	54	10
----	---	----	----

41	27	54	3	0
----	----	----	---	---

72	4
----	---

54	10
----	----

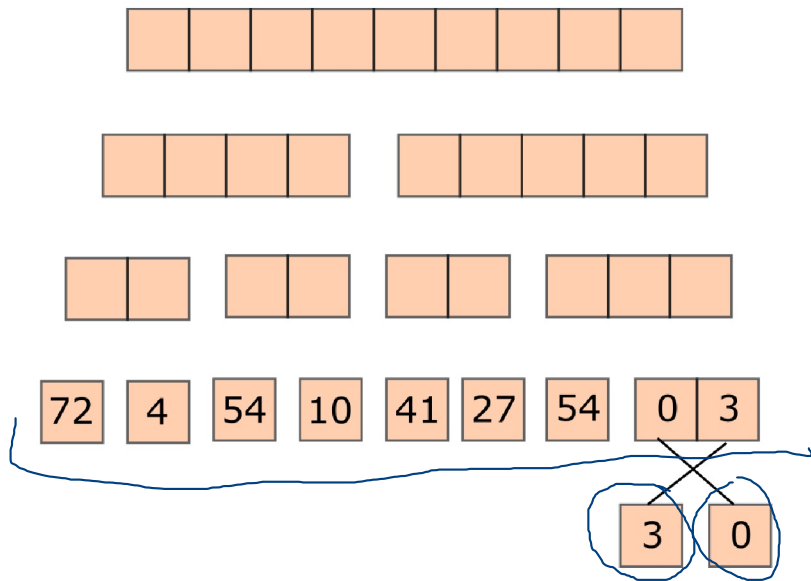
41	27
----	----

54	3	0
----	---	---

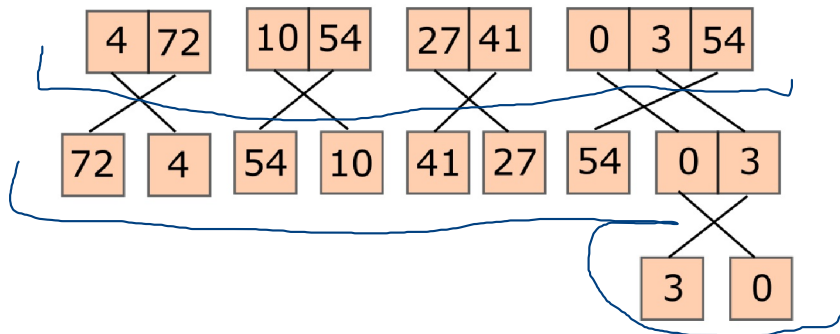
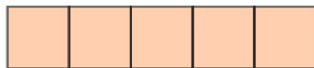
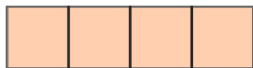
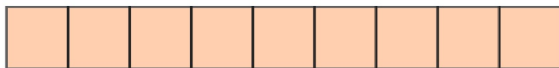
72	4	54	10	41	27	54	3	0
----	---	----	----	----	----	----	---	---

3	0
---	---

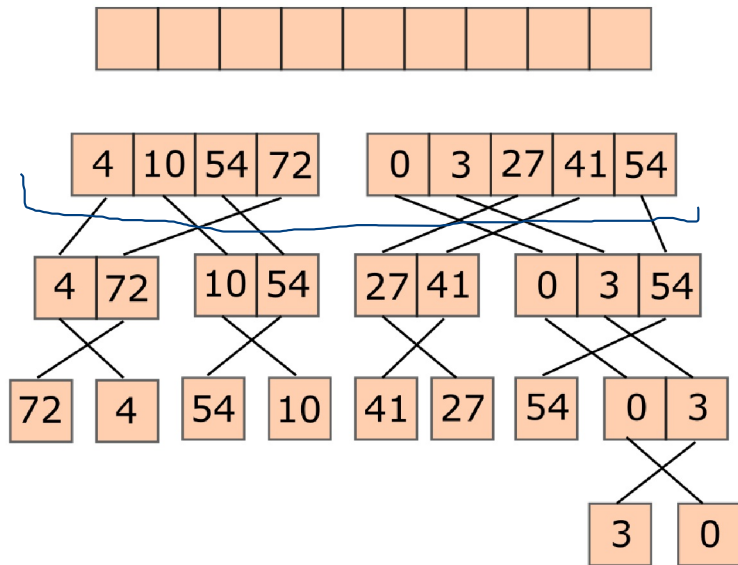
Сортировка слиянием



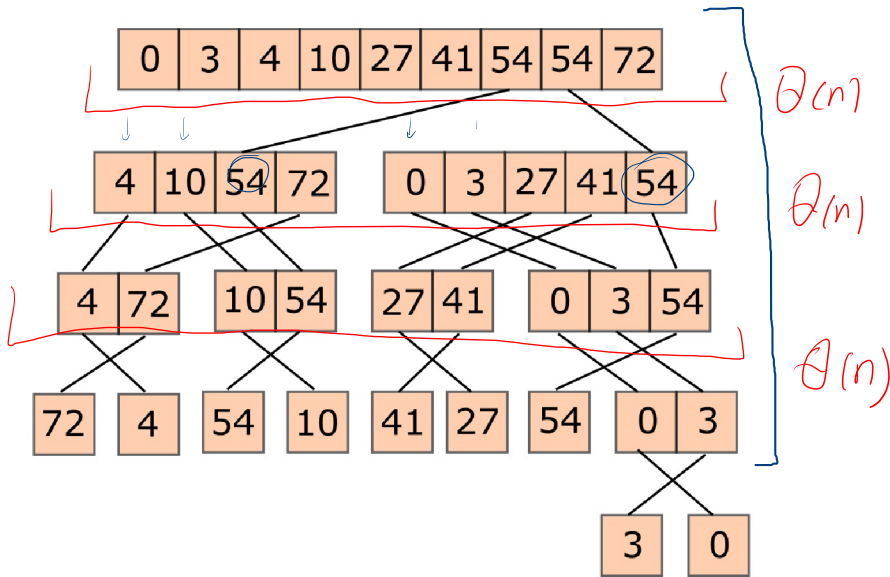
Сортировка слиянием



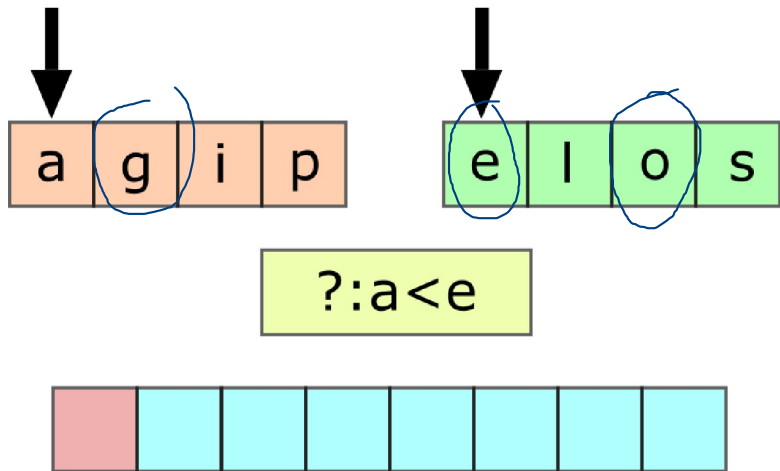
Сортировка слиянием



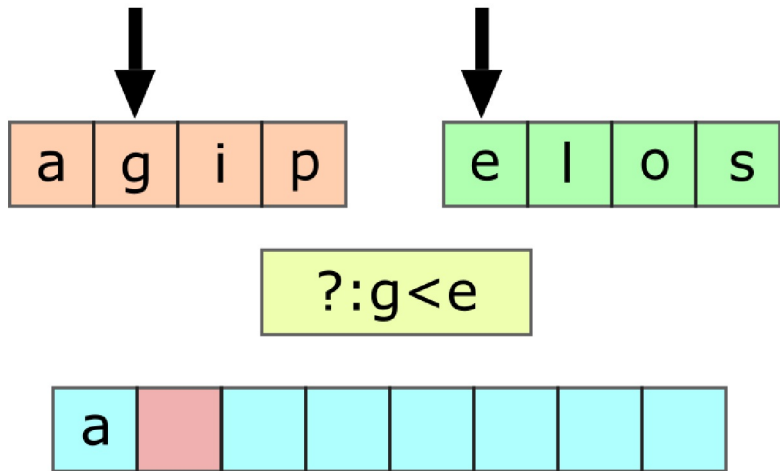
Сортировка слиянием



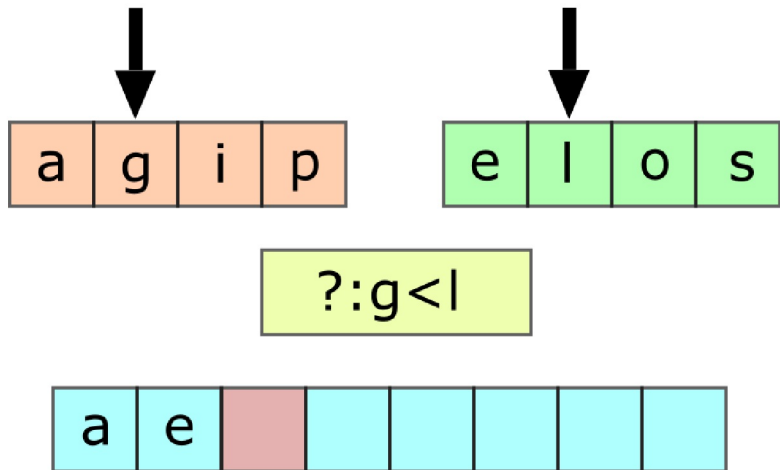
Merging



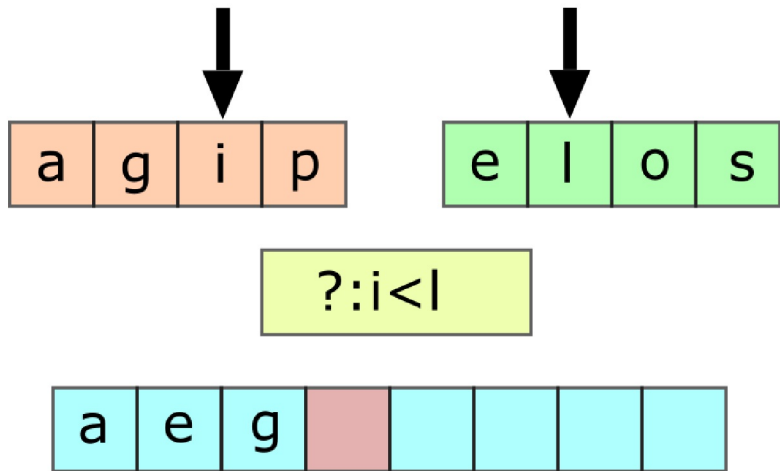
Merging



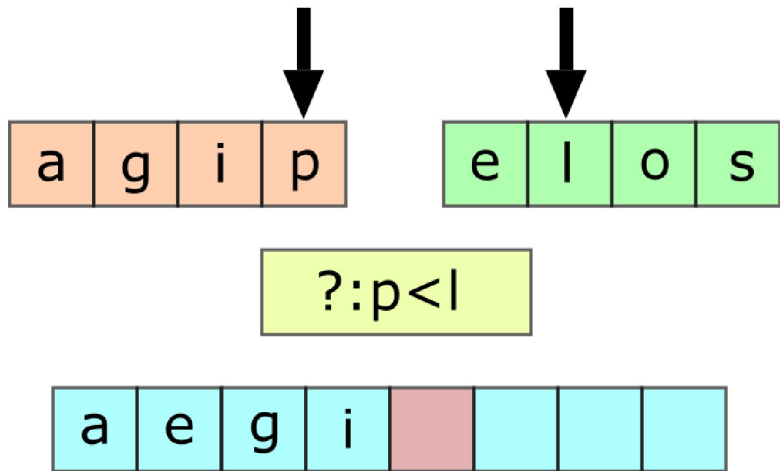
Merging



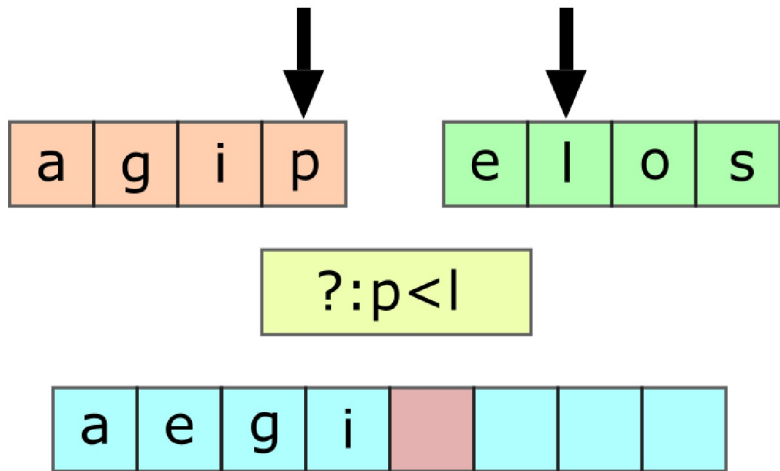
Merging



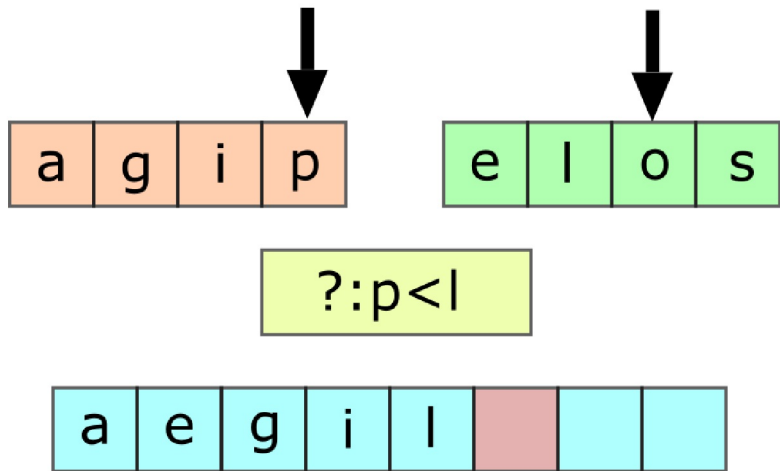
Merging



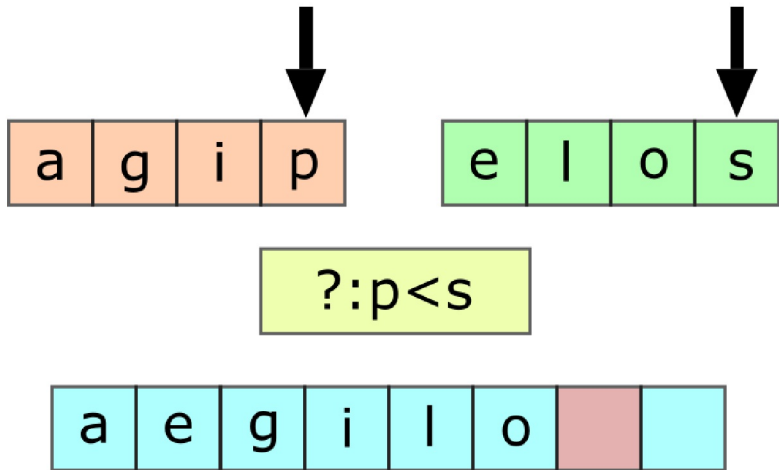
Merging



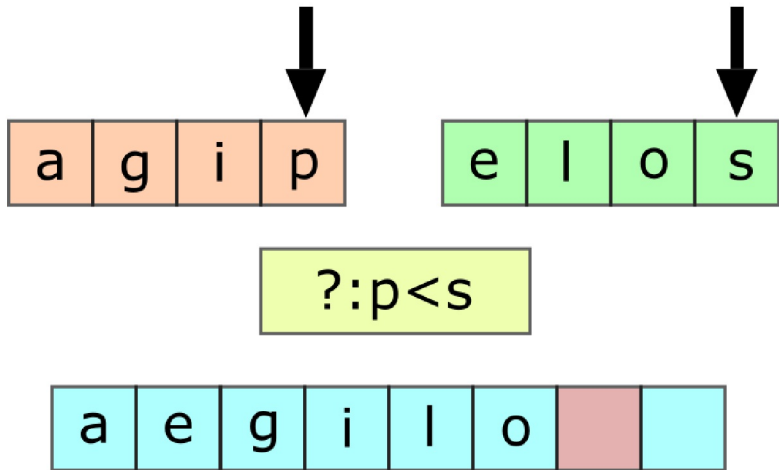
Merging



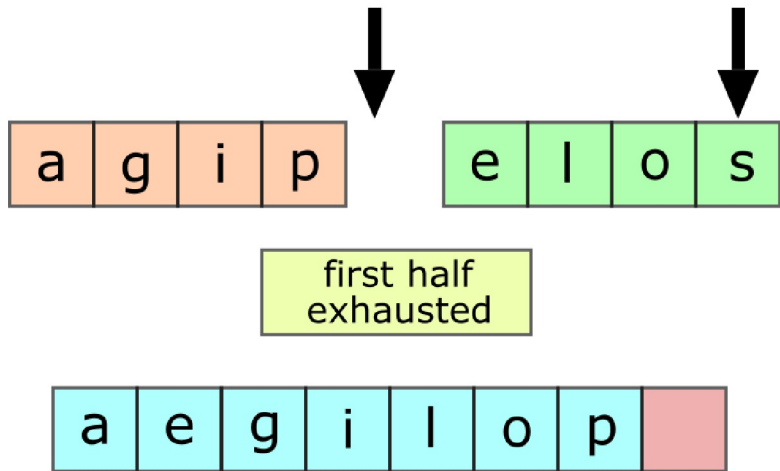
Merging



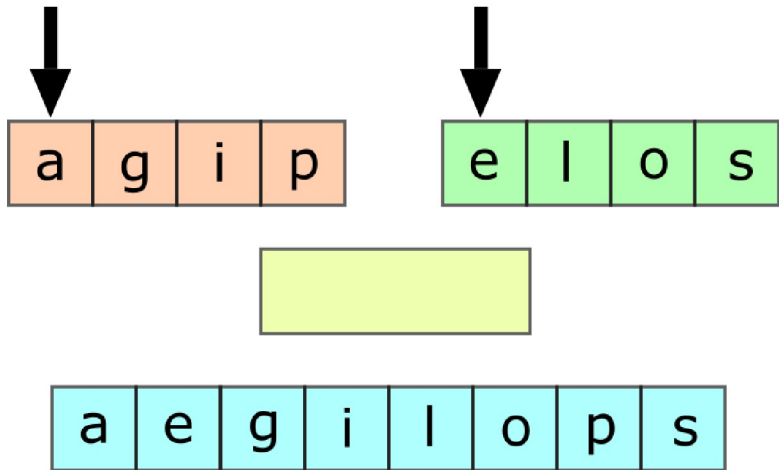
Merging



Merging



Merging



Анализ операции слияния

MERGE(A, p, q, r)

1 $n_1 = q - p + 1$

2 $n_2 = r - q$

3 Пусть $L[1..n_1 + 1]$ и $R[1..n_2 + 1]$ — новые массивы

4 **for** $i = 1$ **to** n_1

5 $L[i] = A[p + i - 1]$

6 **for** $j = 1$ **to** n_2

7 $R[j] = A[q + j]$

8 $L[n_1 + 1] = \infty$

9 $R[n_2 + 1] = \infty$

10 $i = 1$

11 $j = 1$

12 **for** $k = p$ **to** r

13 **if** $L[i] \leq R[j]$

14 $A[k] = L[i]$

15 $i = i + 1$

16 **else** $A[k] = R[j]$

17 $j = j + 1$



копирование

конец строки, можно
проверять как угодно

запись в исходный массив
упорядоченных данных

Оценка времени работы Merge Sort (рекуррентная формула)

MERGE-SORT(A, p, r)

1 if $p < r$

2 $q = \lfloor (p+r)/2 \rfloor$

3 MERGE-SORT(A, p, q)

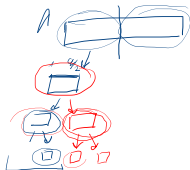
4 MERGE-SORT($A, q+1, r$)

5 MERGE(A, p, q, r)

$T(n)$

$T(\frac{n}{2})$

$\Theta(n)$



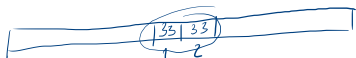
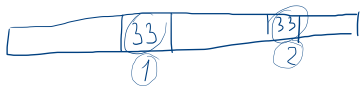
$$T(n) = \begin{cases} \Theta(1) & n=1 \\ 2T(\frac{n}{2}) + \Theta(n) & n>1 \end{cases}$$

$$\begin{aligned} T(n) &= 2T(\frac{n}{2}) + \Theta(n) = 2(2T(\frac{n}{4}) + \Theta(\frac{n}{2})) + \Theta(n) = \\ &= 4T(\frac{n}{4}) + \underbrace{2\Theta(\frac{n}{2})}_{\Theta(n)} + \Theta(n) = \underbrace{8T(\frac{n}{8}) + 3\Theta(n)}_{2^3} = \end{aligned}$$

$$\underbrace{\frac{n}{2^k} = 1}_{k = \log n}$$

$$\begin{aligned} &= 2^{\log n} \underbrace{T(1)}_1 + \log n \cdot n \cdot c' = \underbrace{n \cdot c + n \log n \cdot c'}_{\Theta(n \log n)} = \\ &\quad \Theta(n \log n) \end{aligned}$$

Про устойчивость алгоритмов



$n \log n$

[Merge Sort
Q-sort

устойчив
неустойчив

