# Лекция по курсу «Алгоритмы и структуры данных» / «Технологии и методы программирования»

## Методы сортировки

Мясников Е.В.

# Задача сортировки

Пусть имеется **N** записей:

$R_1, R_2, \ldots, R_N$

Каждая запись $R_i$ имеет ключ $K_i$ и, возможно, дополнительную информацию.

Для ключей вводится отношение порядка так что для любых трех ключей **a,b,c:**

1) справедливо ровно одно из соотношений: **a < b, a > b, a = b**
   (трихотомия)
2) если (**a < b**) и (**b < c**), то (**a < c**) (транзитивность)

**Задача сортировки:**

Найти такую перестановку записей

$R_{P(1)}, R_{P(2)}, \ldots, R_{P(N)}$

при которой ключи располагаются в порядке неубывания:

$K_{P(1)} <= K_{P(2)} <= \ldots <= K_{P(N)}$

Методы сортировки:

внутренние – все записи хранятся в оперативной памяти

внешние – записи хранятся во внешней памяти и целиком не могут быть размещены в оперативной

# Методы внутренней сортировки

**Сортировка вставками (включением)**

- **Метод простого включения**
- **Метод включения с бинарными вставками**
- **Метод Шелла (включения с уменьшающимися расстояниями)**

**Сортировка простого выбора (извлечения)**

**Сортировка обменами**

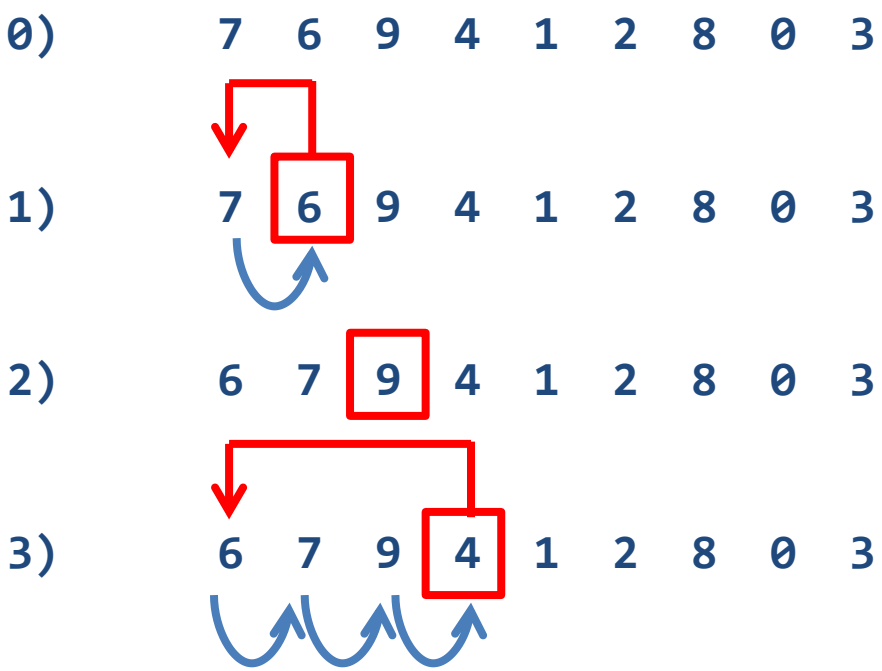
- **Простая обменная сортировка (методы пузырька)**
- **Шейкерная сортировка**
- **Быстрая сортировка (QuickSort)**

**Внутренние сортировки слиянием**

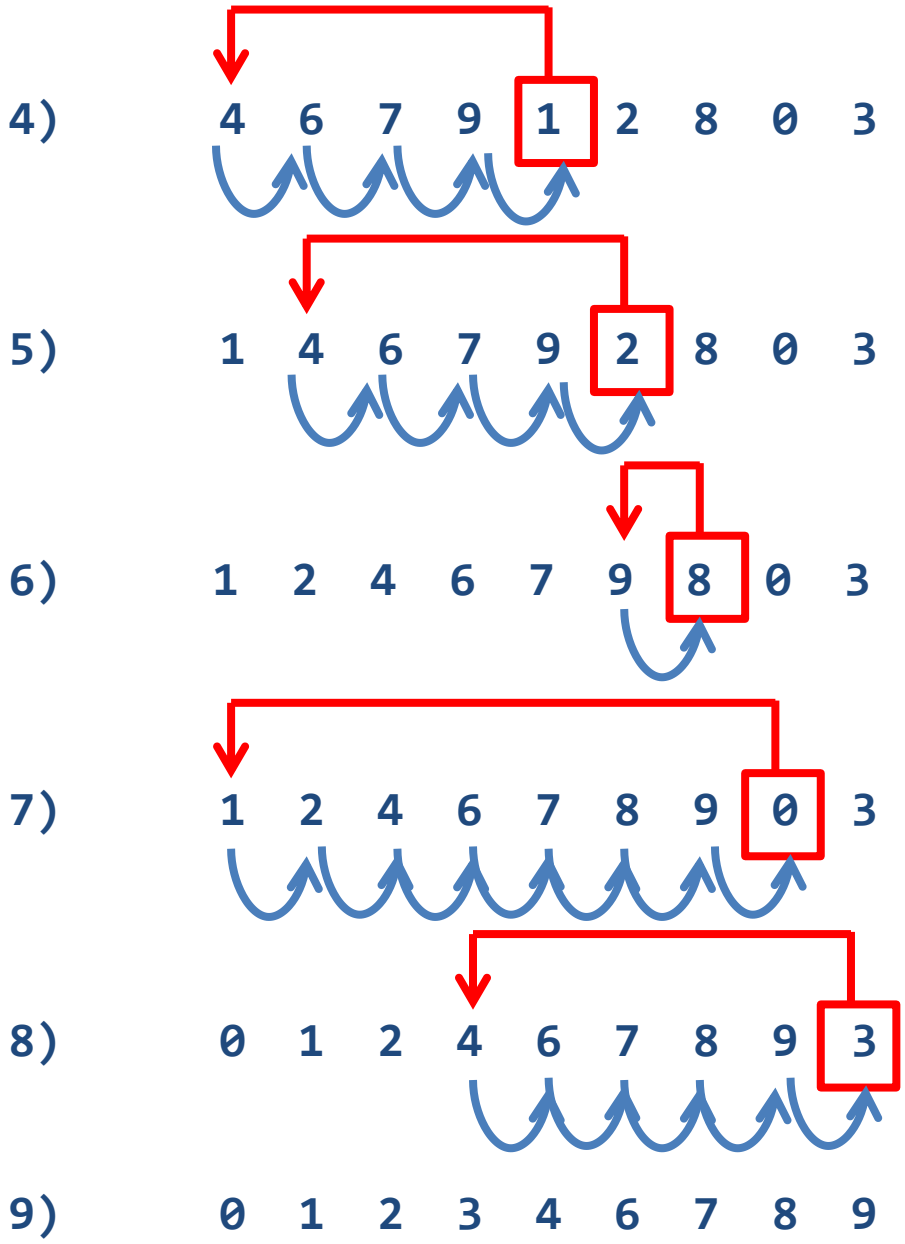
- **Естественное двухпутевое слияние**
- **Простое двухпутевое слияние**

# Метод простого включения

Пусть имеется массив ключей $K_1$, $K_2$, ..., $K_n$. Для каждого элемента массива, начиная со второго, производится сравнение с элементами с меньшим индексом (элемент $K_i$ последовательно сравнивается с элементами $K_{(i-1)}$, $K_{(i-2)}$, …) и до тех пор, пока для очередного элемента $K_j$ выполняется соотношение $K_j > K_i$, $K_i$ и $K_j$ меняются местами. Если удается встретить такой элемент $K_j$, что $K_j <= K_i$, или если достигнута нижняя граница массива, производится переход к обработке элемента $K_{(i+1)}$ (пока не будет достигнута верхняя граница массива).



# Метод простого включения



# Метод простого включения

В лучшем случае (когда массив уже упорядочен) для выполнения алгоритма с массивом из n элементов потребуется n-1 сравнение и 0 пересылок. В худшем случае (когда массив упорядочен в обратном порядке) потребуется n*(n-1)/2 сравнений и столько же пересылок. Таким образом, можно оценивать сложность метода простых включений как $O(n^2)$.

# Метод бинарных вставок

Можно сократить число сравнений, применяемых в методе простых включений, если воспользоваться тем фактом, что при обработке элемента Ki массива элементы K1, K2, ..., K(i-1) уже упорядочены, и воспользоваться для поиска элемента, с которым должна быть произведена перестановка, методом двоичного деления.
В этом случае оценка числа требуемых сравнений становится O(n*log n).
Заметим, что поскольку при выполнении перестановки требуется сдвижка на один элемент нескольких элементов, то оценка числа пересылок остается $O(n^2)$.
Это называется методом бинарных вставок.

# Метод Шелла

На самостоятельное изучение!

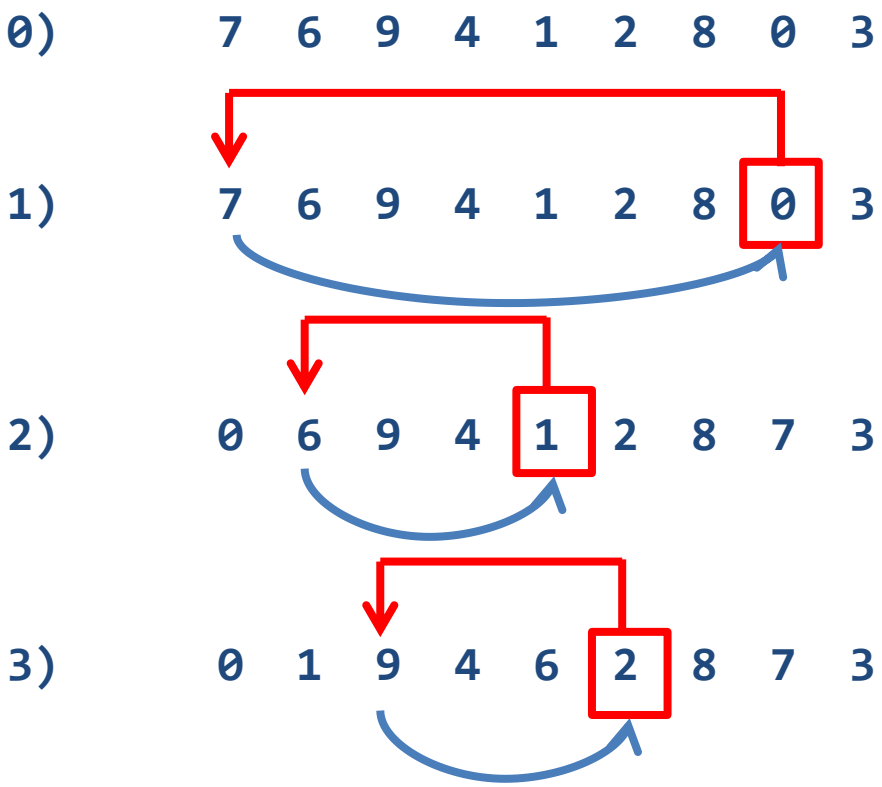
Ссылки:

- Д. Кнут. Искусство программирования. Том 3. Сортировка и поиск
- **https://www.youtube.com/watch?v=CmPA7zE8mx0**

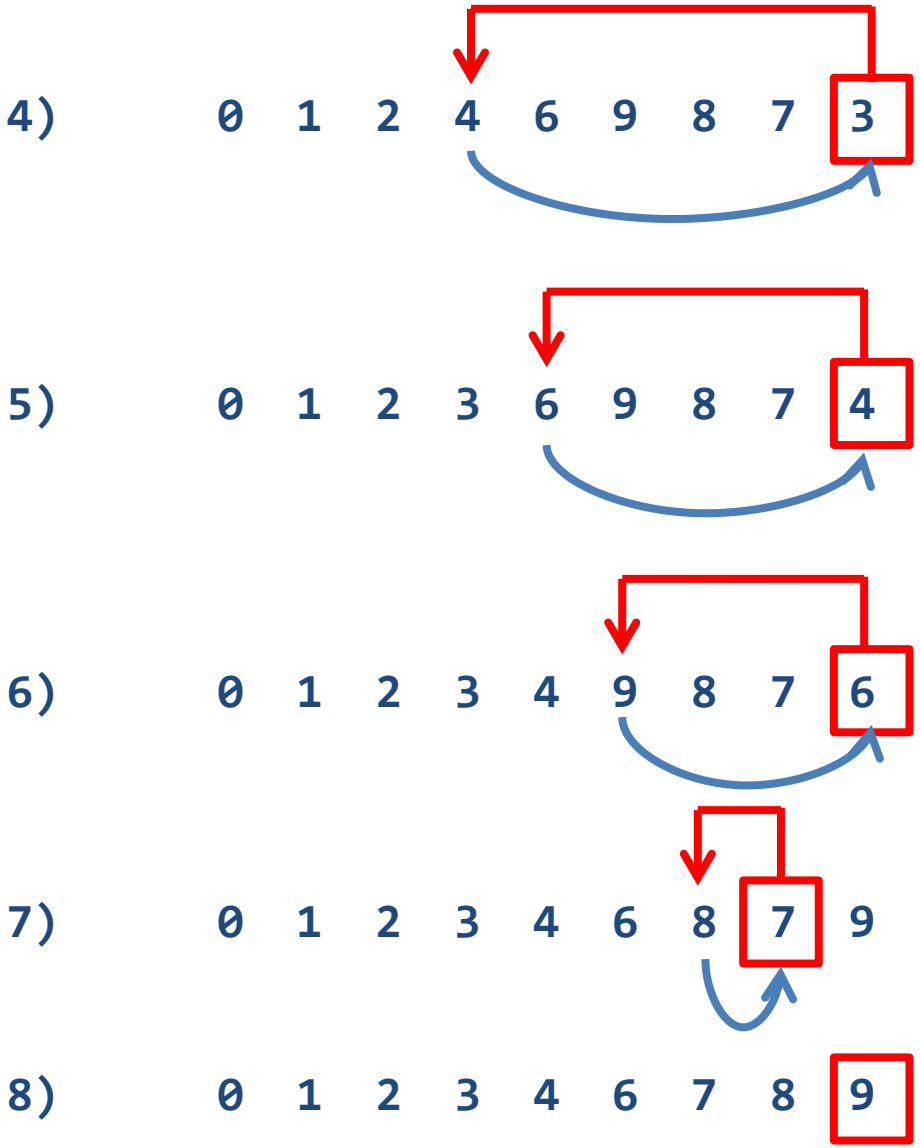
Hungarian folk dance

# Сортировка простым выбором (извлечением)

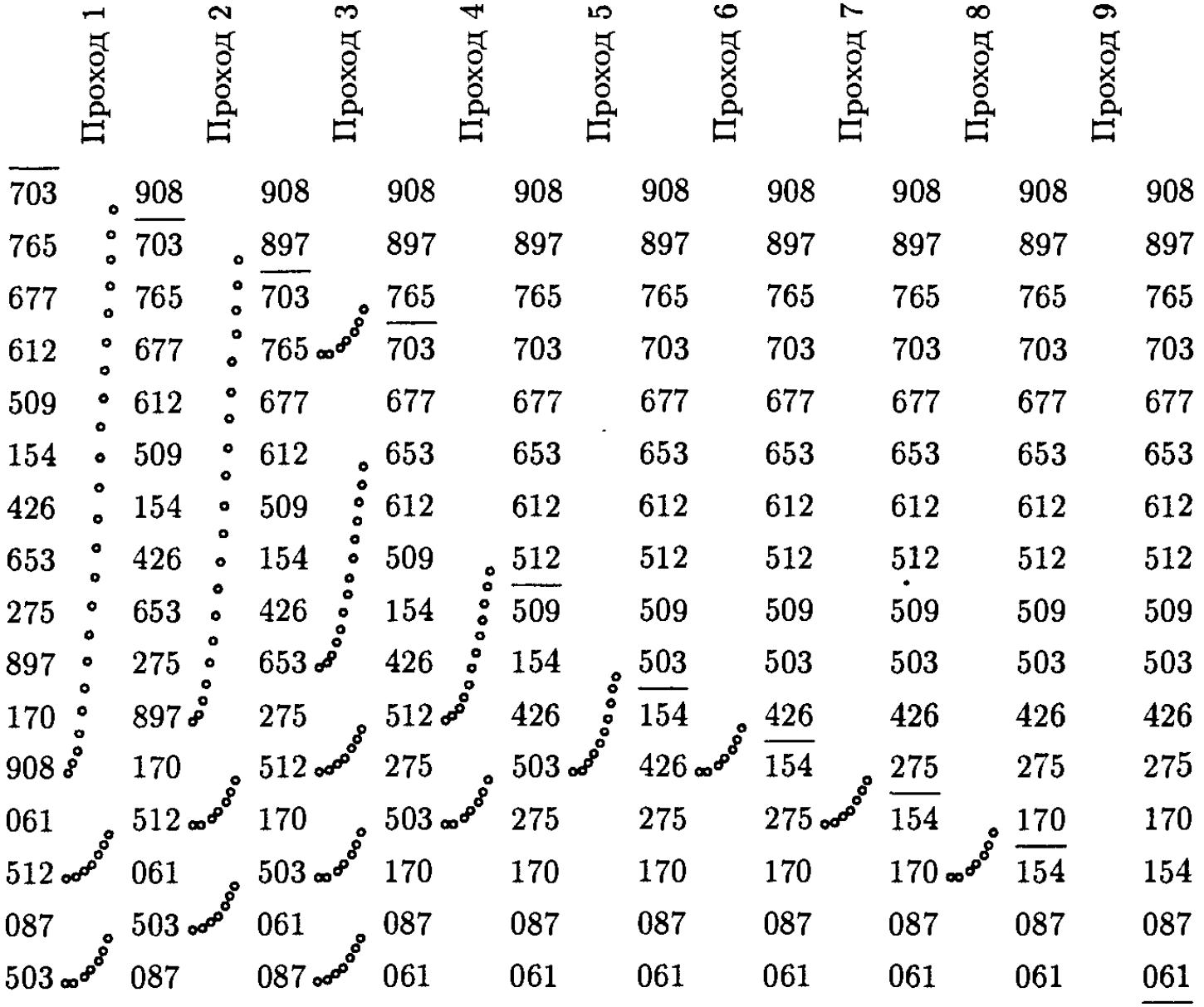
При сортировке массива a[1], a[2], ..., a[n] методом простого выбора среди всех элементов находится элемент с наименьшим значением a[i], и a[1] и a[i] обмениваются значениями. Затем этот процесс повторяется для получаемых подмассивов a[2], a[3], ..., a[n], ... a[j], a[j+1], ..., a[n] до тех пор, пока мы не дойдем до подмассива a[n], содержащего к этому моменту наибольшее значение.



# Сортировка простым выбором (извлечением)

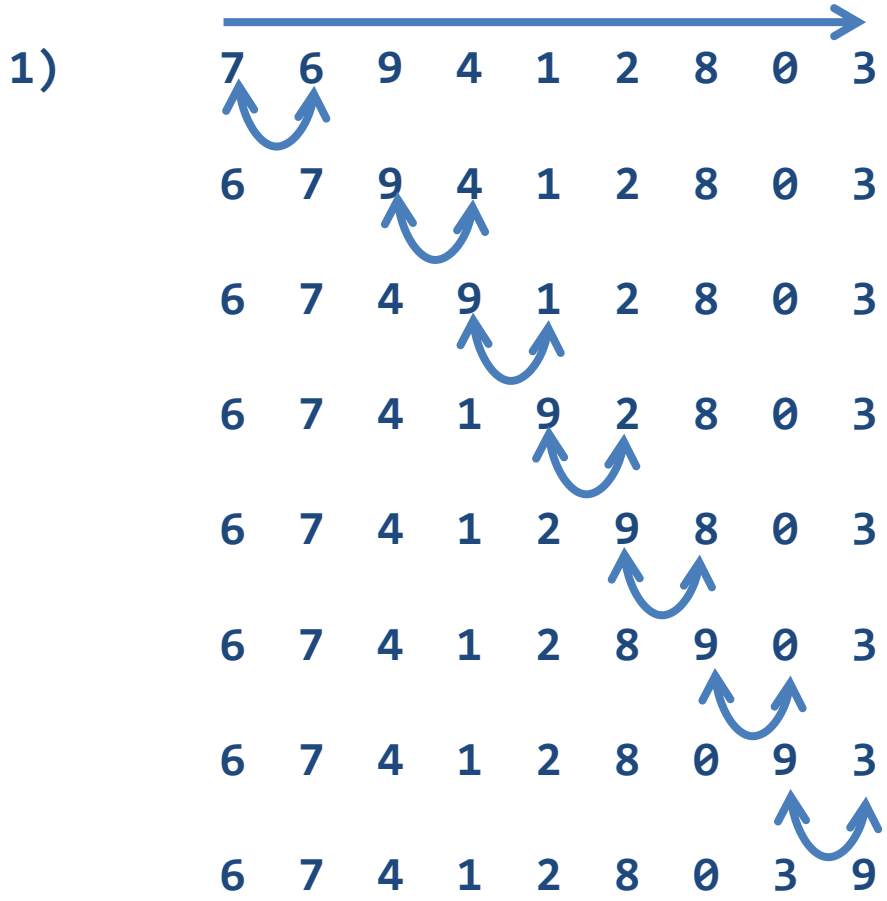


# Простая обменная сортировка (пузырек)

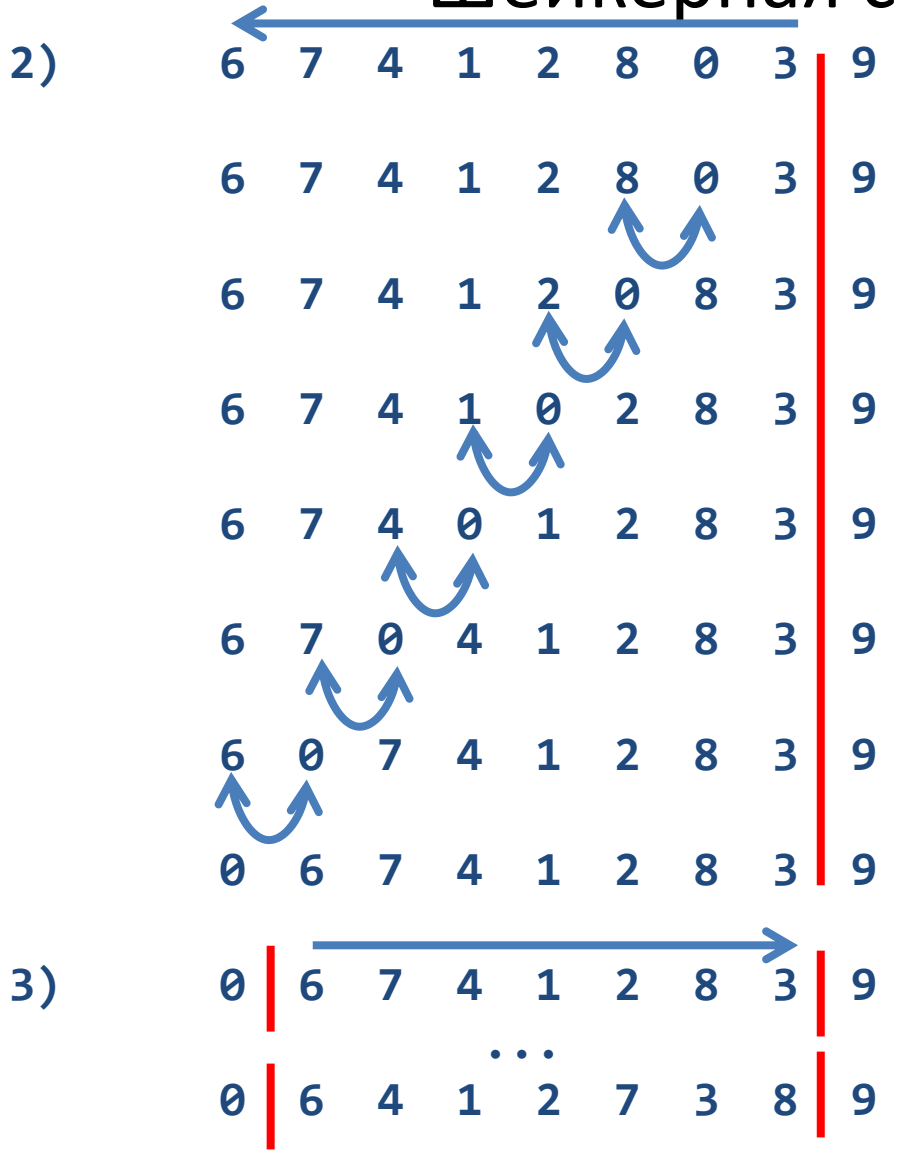


# Шейкерная сортировка

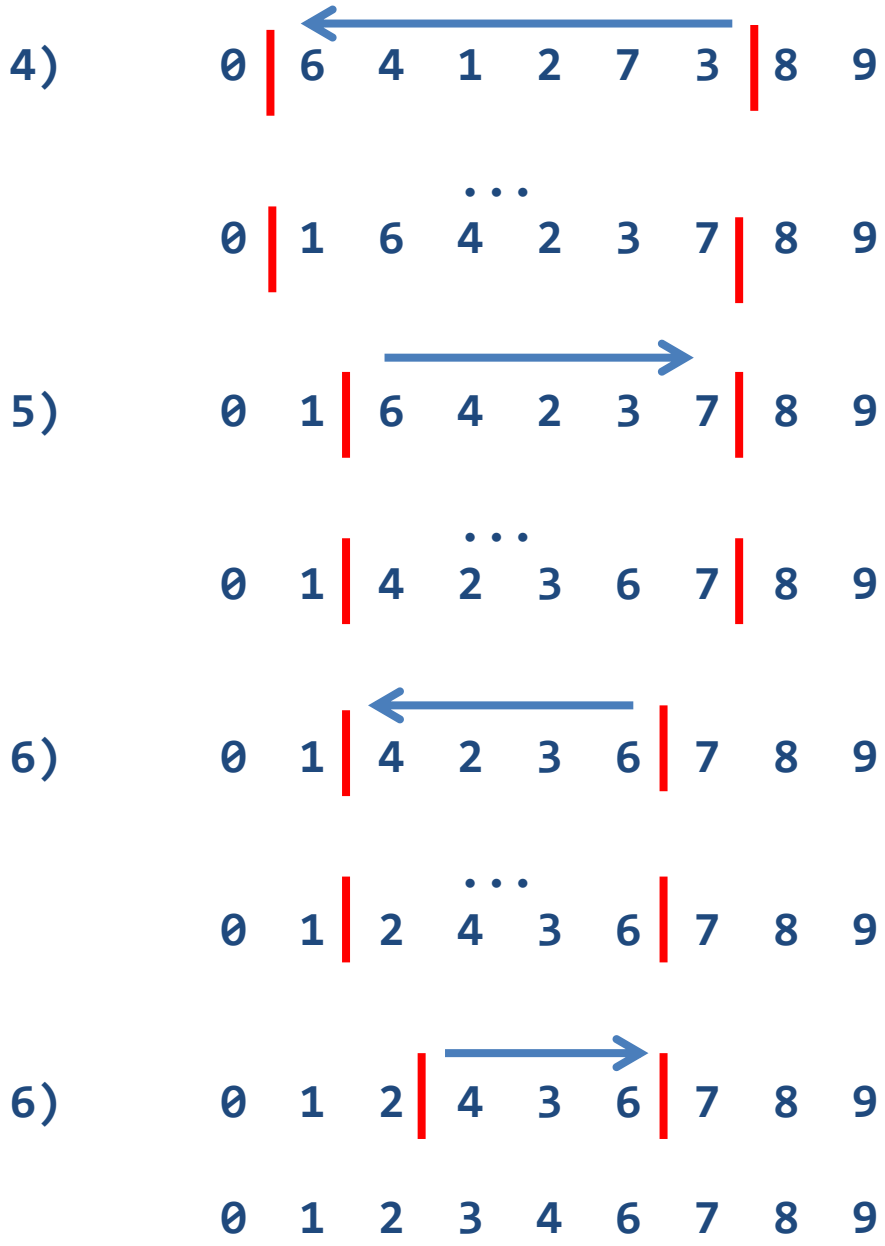
Последовательность записей просматривается попеременно в обоих направлениях.
На четной итерации всплывает самый легкий элемент,
на нечетной опускается самой тяжелый.



# Шейкерная сортировка

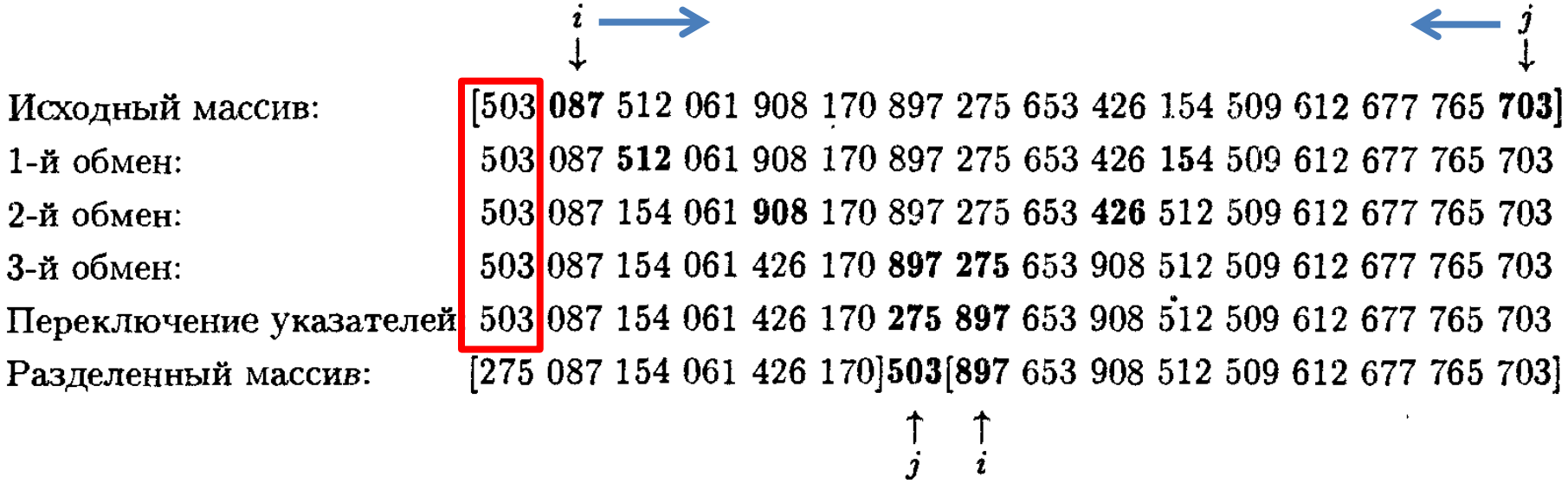


# Шейкерная сортировка



# Быстрая сортировка (QuickSort, сортировка Хоара)

Основная идея алгоритма состоит в том, чтобы на каждом шаге ставить один из элементов массива (опорный) на место, где он должен будет находиться после сортировки.
Для этого необходимо определить элементы, которые должны быть слева (и справа) от него.
В алгоритме быстрой сортировки одновременно происходит и перекомпоновка массива так, что после нее слева оказываются элементы меньшие опорного, а справа – большие.
После этого массив разбивается на два подмассива меньшей длины и сортируются независимо друг от друга.



# Сортировки слияниями

**Слияние:**

Пусть имеются два отсортированных в порядке возрастания массива

$p_1, p_2, \ldots, p_n$

и

$q_1, q_2, \ldots, q_m$

и массив для сохранения результата

$r_1, r_2, \ldots, r_{n+m}$

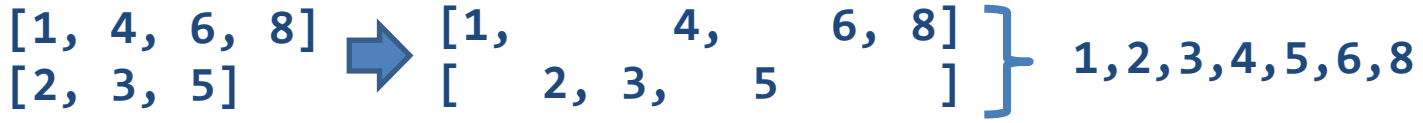
**Процедура слияния:**

Указатели (индексы) $i$ и $j$ передвигаются последовательно по элементам упорядоченных массивов $p$ и $q$.

Элементы $p_i$ и $q_j$ сравниваются и меньшее из значений записывается в очередной элемент массива-результата $r_{i+j}$.

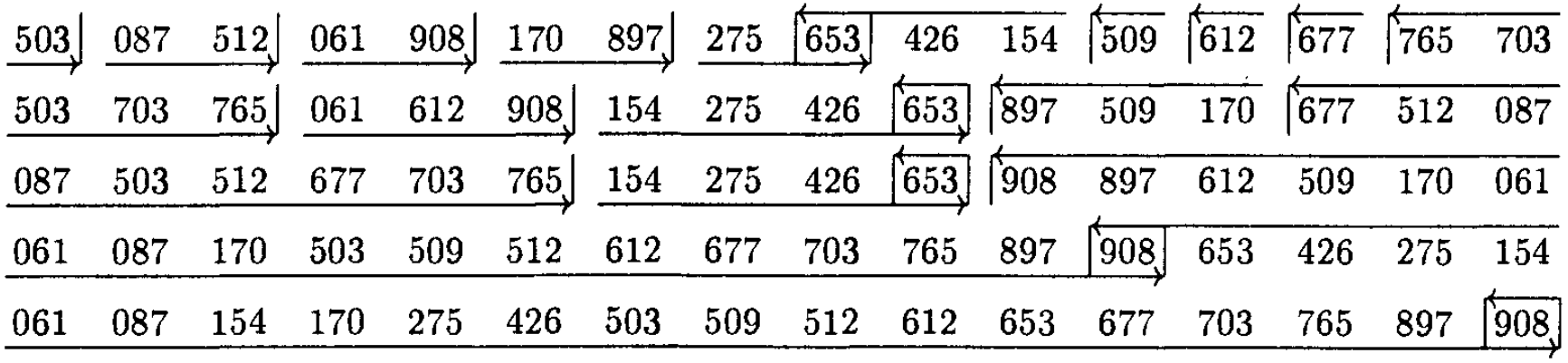
Указатель (или индекс) элемента сдвигается на 1.

В том случае, если один из массивов исчерпан, остаток второго массива переписывается в массив- результат.



# Естественное двухпутевое слияние

Основная идея алгоритма состоит в поиске двух возрастающих последовательностей: одной с начала массива (при просмотре слева направо), другой – с конца массива, (при просмотре справа налево).
Выполняется слияние найденных последовательностей с записью во вспомогательный массив поочередно слева направо (в начало) и справа налево (в конец массива).
После исчерпания всех элементов исходный и вспомогательный массив меняются местами.



# Простое двухпутевое слияние

Длины серий устанавливаются принудительно и равны $2^k$, где $k$ – номер итерации: 1 на 1 шаге, 2 на втором, 4 на третьем и т.д.
Преимущество: нет проверки конца серий,
недостаток: длинные серии обрабатываются медленнее.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 503 | 087 | 512 | 061 | 908 | 170 | 897 | 275 | 653 | 426 | 154 | 509 | 612 | 677 | 765 | 703 |
| 503 | 703 | 512 | 677 | 509 | 908 | 426 | 897 | 653 | 275 | 170 | 154 | 612 | 061 | 765 | 087 |
| 087 | 503 | 703 | 765 | 154 | 170 | 509 | 908 | 897 | 653 | 426 | 275 | 677 | 612 | 512 | 061 |
| 061 | 087 | 503 | 512 | 612 | 677 | 703 | 765 | 908 | 897 | 653 | 509 | 426 | 275 | 170 | 154 |
| 061 | 087 | 154 | 170 | 275 | 426 | 503 | 509 | 512 | 612 | 653 | 677 | 703 | 765 | 897 | 908 |

# Оценки методов сортировки

| Сортировка | Время | | | Память |
|---|---|---|---|---|
| | лучшее | среднее | худшее | |
| Выбором | $O(n^2)$ | $O(n^2)$ | $O(n^2)$ | O(1) |
| Вставками | O(n) | $O(n^2)$ | $O(n^2)$ | O(1) |
| Шелла | Зависит от шага<br>O(n log(n)) | Зависит от шага<br>$O(n^{5/3})$, $O(n^{7/6})$,... | Зависит от шага<br>$O(n^2)$, $O(n^{4/3})$,… | O(1) |
| Пузырьком | O(n) | $O(n^2)$ | $O(n^2)$ | O(1) |
| Шейкерная | O(n) | $O(n^2)$ | $O(n^2)$ | O(1) |
| Быстрая | O(n log(n)) | O(n log(n)) | $O(n^2)$<br>*маловероятно* | O(log(n))<br>Почему? |
| Слияниями | O(n log(n)) | O(n log(n)) | O(n log(n)) | O(n) |

# Внешние сортировки слиянием

# Внешние сортировки слиянием

Пусть имеется 5 млн. записей.
Разобьём весь набор на 5 частей, отсортируем их в оперативной памяти, и запишем на носители поочередно.

| | |
|---|---|
| Лента 1 | $R_1 \dots R_{1000000}$; $R_{2000001} \dots R_{3000000}$; $R_{4000001} \dots R_{5000000}$ |
| Лента 2 | $R_{1000001} \dots R_{2000000}$; $R_{3000001} \dots R_{4000000}$ |
| Лента 3 | (Пустая) |
| Лента 4 | (Пустая) |

Выполним первый этап слияния:

| | |
|---|---|
| Лента 3 | $R_1 \dots R_{2000000}$; $R_{4000001} \dots R_{5000000}$ |
| Лента 4 | $R_{2000001} \dots R_{4000000}$ |

Выполним второй этап слияния:

| | |
|---|---|
| Лента 1 | $R_1 \dots R_{4000000}$ |
| Лента 2 | $R_{4000001} \dots R_{5000000}$ |

# Сбалансированное слияние (1/2)

1. Распределить начальные серии попеременно на ленты T1 и T2.
2. Выполнить слияние серий с лент T1 и T2 на T3, затем остановиться, если T3 содержит только одну серию.
3. Скопировать серии с T3 попеременно на T1 и T2, затем вернуться к шагу B2.

1) **Лента 1: R1….R1000000, R2000001…R3000000 , R4000001…R5000000**
   **Лента 2: R1000001….R2000000, R3000001…R4000000**
   Лента 3: (пустая)

2) Лента 1: R1….R1000000, R2000001…R3000000 , R4000001…R5000000
   Лента 2: R1000001….R2000000, R3000001…R4000000
   **Лента 3: R1….R2000000, R2000001…R4000000 , R4000001…R5000000**

3) **Лента 1: R1….R2000000, R4000001…R5000000**
   **Лента 2: R2000001….R4000000**
   Лента 3: R1….R2000000, R2000001…R4000000 , R4000001…R5000000

4) Лента 1: R1….R2000000, R4000001…R5000000
   Лента 2: R2000001….R4000000
   **Лента 3: R1….R4000000, R4000001…R5000000**

# Сбалансированное слияние (2/2)

1. Распределить начальные серии попеременно на ленты T1 и T2.
2. Выполнить слияние серий с лент T1 и T2 на T3, затем остановиться, если T3 содержит только одну серию.
3. Скопировать серии с T3 попеременно на T1 и T2, затем вернуться к шагу B2.

4) Лента 1: R1….R2000000, R4000001…R5000000
Лента 2: R2000001….R4000000
**Лента 3: R1….R4000000, R4000001…R5000000**

**5) Лента 1: R1….R4000000**
**Лента 2: R4000001….R5000000**
Лента 3: R1….R4000000, R4000001…R5000000

6) Лента 1: R1….R4000000
Лента 2: R4000001….R5000000
**Лента 3: R1…R5000000**

**Недостаток:** шаг 3 – просто копирование без слияний

# Двухфазное слияние

**A1.** Распределить начальные серии попеременно на ленты T1 и T2.

**A2.** Слить серии с лент T1 и T2 на T3; остановиться, если на T3 содержится только одна серия.

**A3.** Скопировать *половину* серий T3 на T1.

**A4.** Слить серии с лент T1 и T3 на T2; остановиться, если на T2 содержится только одна серия.

**A5.** Скопировать *половину* серий с T2 на T1. Вернуться к шагу A2. ▮

1-A1) **Лента 1: R1….R1000000, R2000001…R3000000 , R4000001…R5000000**
**Лента 2: R1000001….R2000000, R3000001…R4000000**
Лента 3: (пустая)

2-A2) Лента 1: R1….R1000000, R2000001…R3000000 , R4000001…R5000000
Лента 2: R1000001….R2000000, R3000001…R4000000
**Лента 3: R1….R2000000, R2000001…R4000000 , R4000001…R5000000**

3-A3) **Лента 1: R1….R2000000**
Лента 2: R1000001….R2000000, R3000001…R4000000
**Лента 3:** R1….R2000000, **R2000001…R4000000 , R4000001…R5000000**

# Двухфазное слияние

**A1.** Распределить начальные серии попеременно на ленты T1 и T2.

**A2.** Слить серии с лент T1 и T2 на T3; остановиться, если на T3 содержится только одна серия.

**A3.** Скопировать *половину* серий T3 на T1.

**A4.** Слить серии с лент T1 и T3 на T2; остановиться, если на T2 содержится только одна серия.

**A5.** Скопировать *половину* серий с T2 на T1. Вернуться к шагу A2. ▮

3-A3) **Лента 1: R1….R2000000**
Лента 2: R1000001….R2000000, R3000001…R4000000
**Лента 3:** R1….R2000000, **R2000001…R4000000 , R4000001…R5000000**

4-A4) Лента 1: R1….R2000000
**Лента 2: R1….R4000000, R4000001…R5000000**
Лента 3: R1….R2000000, R2000001…R4000000 , R4000001…R5000000

4-A5) Лента 1: **R1….R4000000**
Лента 2: R1….R4000000, **R4000001…R5000000**
Лента 3: R1….R2000000, R2000001…R4000000 , R4000001…R5000000

6-A2) **Лента 3: R1….R5000000**