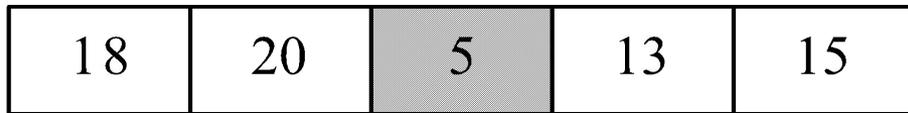


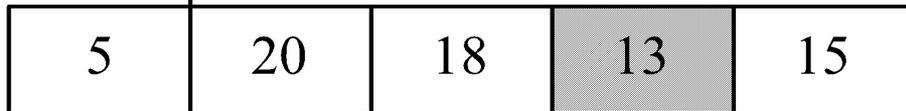
Сортировка посредством выбора

Пример.

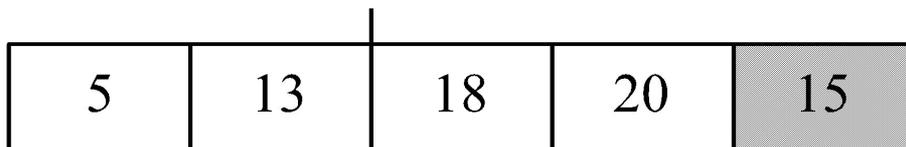
Массив содержит целые числа 18, 20, 5, 13, 15.



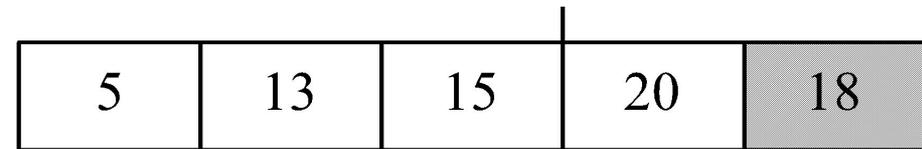
проход 0



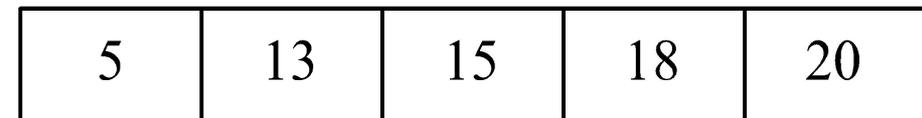
проход 1



проход 2

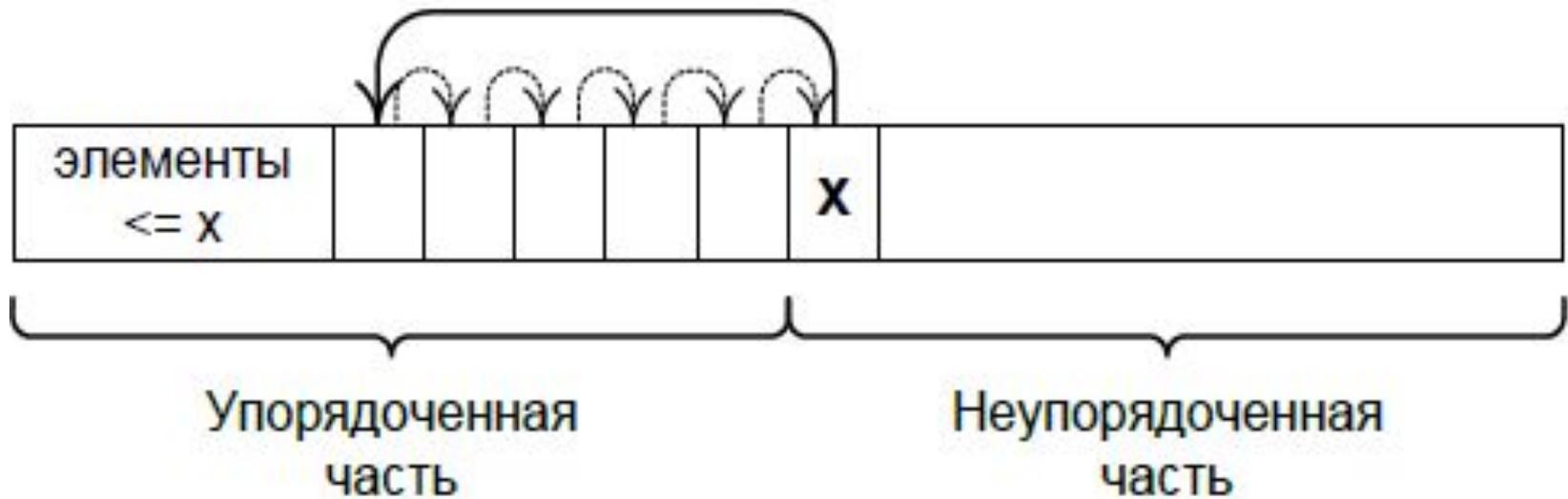


проход 3



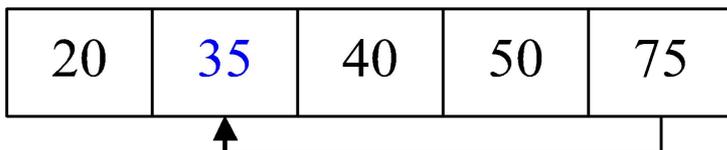
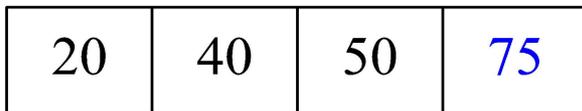
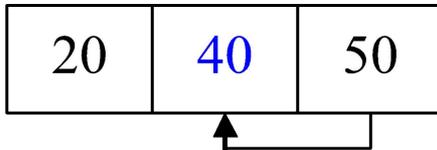
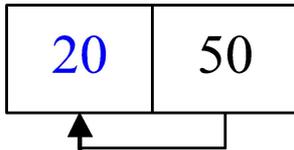
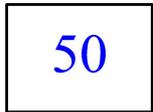
$O(n^2)$

Сортировка вставками



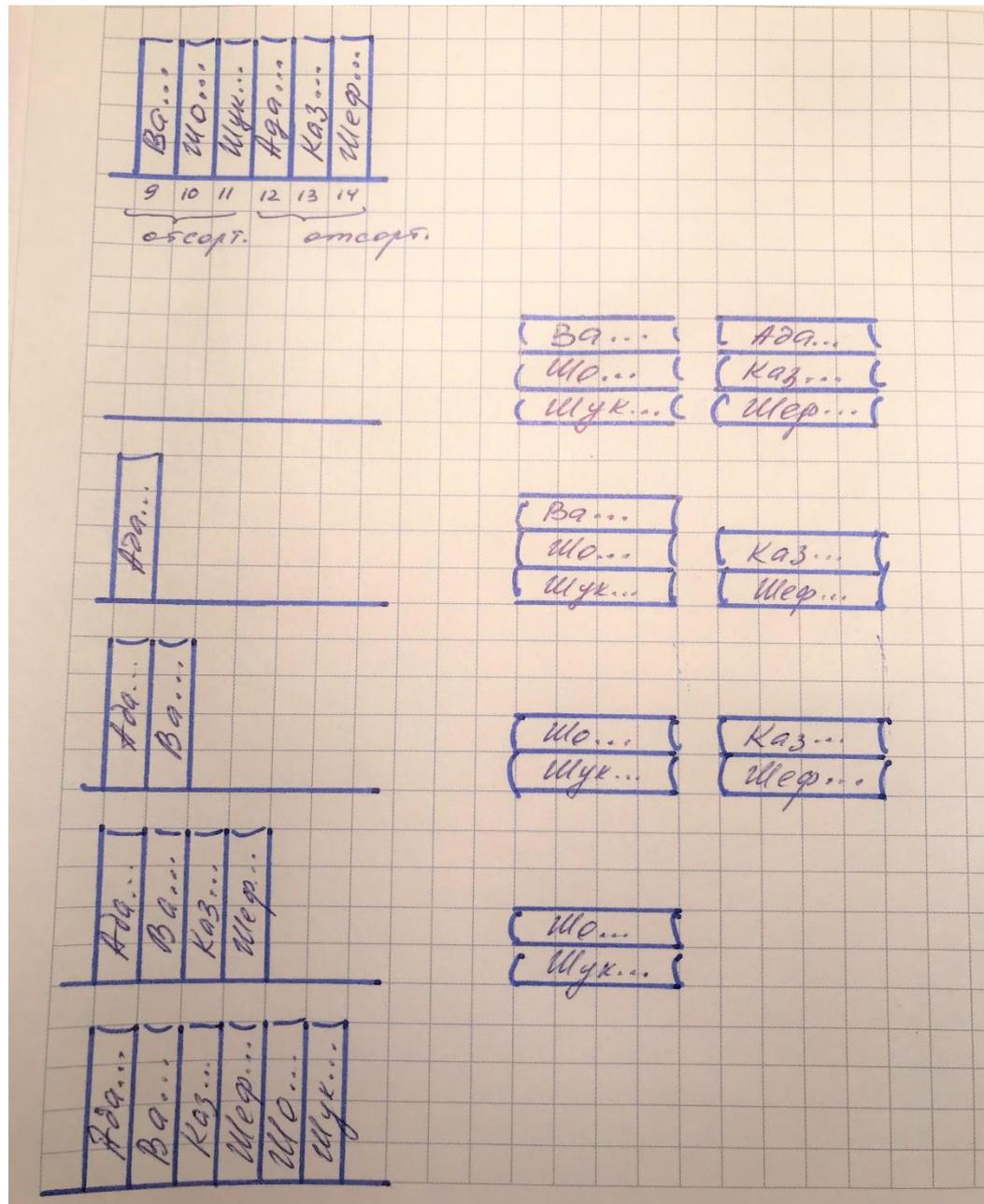
Сортировка вставками

A = 50, 20, 40, 75, 35



$O(n^2)$

Сортировка слиянием (продолжени е)



Сортировка слиянием (продолжение)

- $A[p..q]$ $A[q+1..r]$ $A[p..r]$
- $n1 = q - p + 1$ $A[p..q]$
- $n2 = r - q$ $A[q+1..r]$
- Массив B – $n1$
- Массив C – $n2$
- $A[p..q]$ B
- $A[q+1..r]$ C

Сортировка слиянием (продолжение)

Процедура *Merge*(A, p, q, r)

Вход:

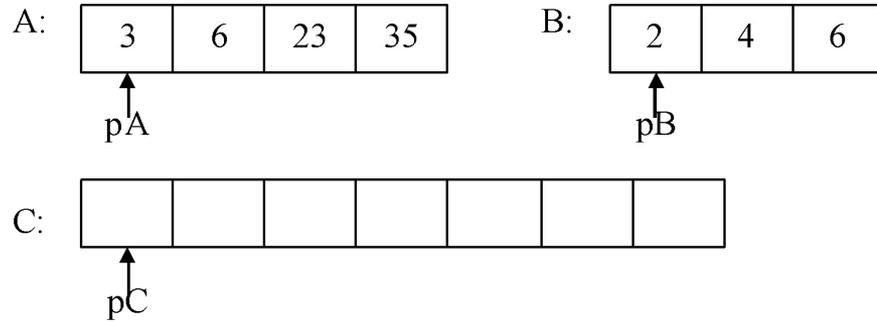
- A : массив
- p, q, r : индексы в массиве A . Подмассивы $A[p..q]$ и $A[q+1..r]$ считаются уже отсортированными.

Результат: подмассив $A[p..r]$, содержащий все элементы, находившиеся в подмассивах $A[p..q]$ и $A[q+1..r]$, но теперь подмассив $A[p..r]$ отсортирован.

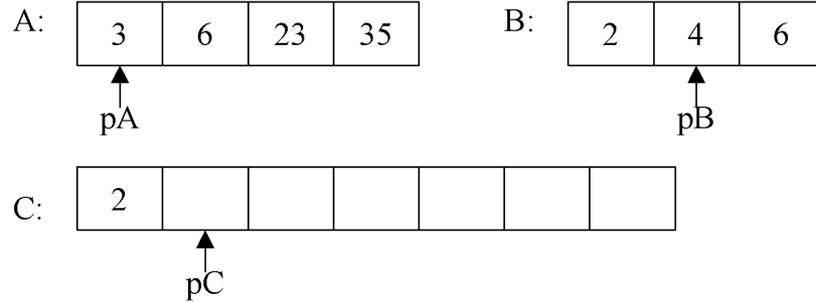
1. Установить n_1 равным $q-p+1$, n_2 – равным $r-q$.
2. $B[1..n_1+1]$ и $C[1..n_2+1]$ представляют собой новые массивы.
3. Скопировать $A[p..q]$ в $B[1..n_1]$, а $A[q+1..r]$ - в $C[1..n_2]$.
4. Установить $B[n_1+1]$ и $C[n_2+1]$ равными «бесконечности».
5. Установить i и j равными 1.
6. Для $k = p$ до r
 - А. Если $B[i] \leq C[j]$, установить $A[k]$ равным $B[i]$ и увеличить i на 1.
 - В. В противном случае ($B[i] > C[j]$) установить $A[k]$ равным $C[j]$ и увеличить j на 1.

Слияние сортированных последовательностей

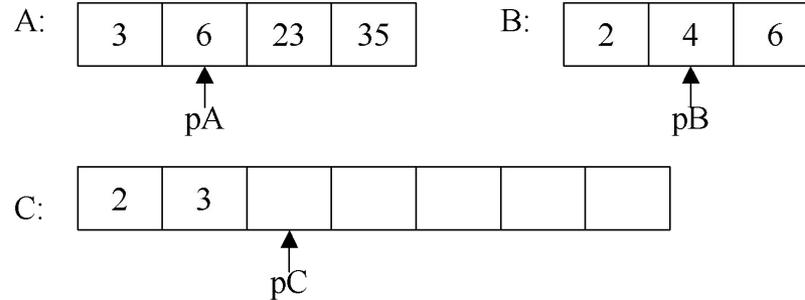
Шаг 1



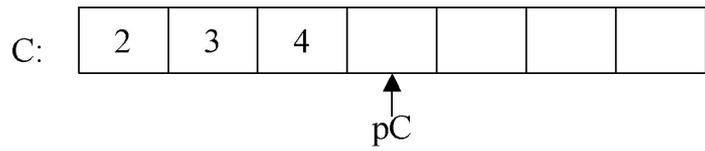
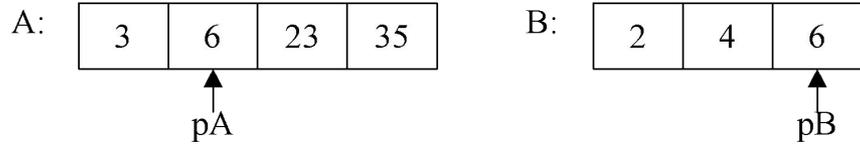
Шаг 2



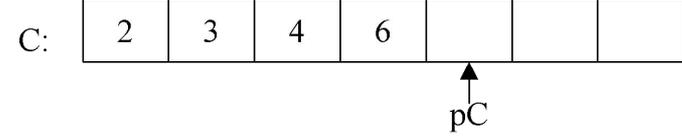
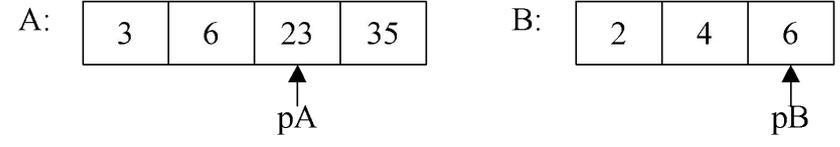
Шаг 3



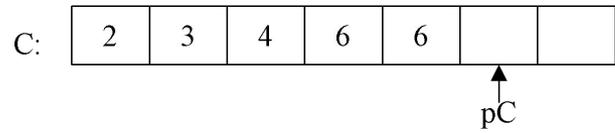
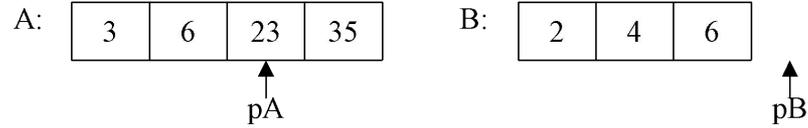
IIIar 4



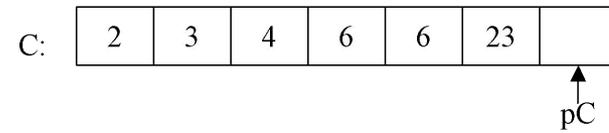
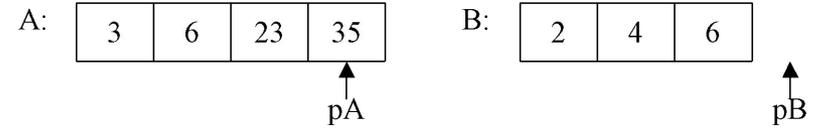
IIIar 5



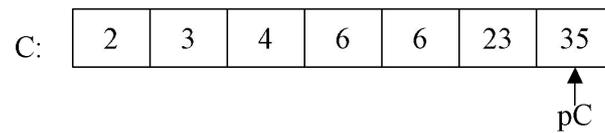
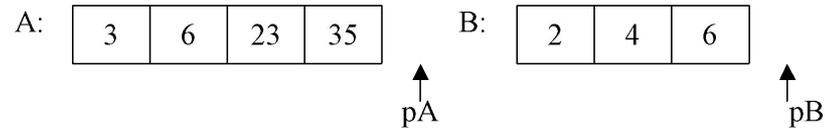
IIIar 6



IIIar 7



IIIar 8



Быстрая сортировка

- Быстрая сортировка работает «на месте», без привлечения дополнительной памяти.
- Асимптотическое время работы быстрой сортировки для среднего случая отличается от времени работы для наихудшего случая.

Парадигма «разделяй и властвуй».

Изначально мы хотим отсортировать все n книг

в слотах с первого до n -го

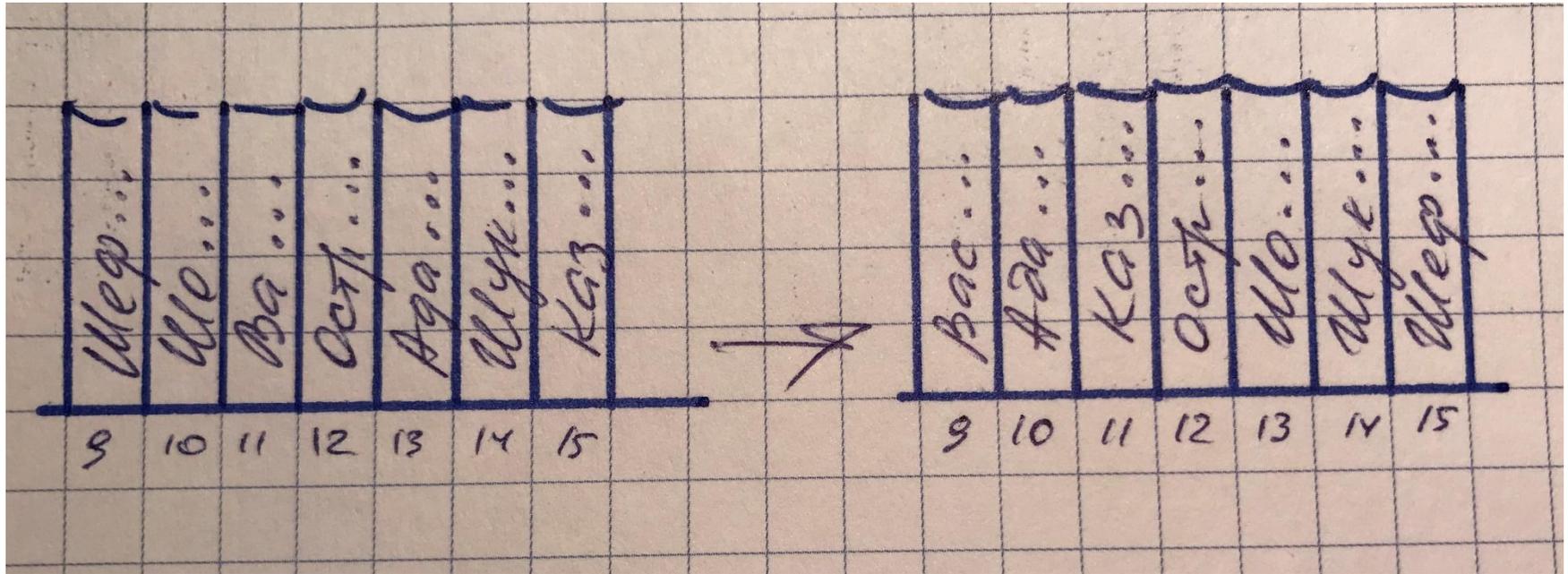
и при этом рассматриваем обобщенную задачу сортировки книг

в слотах с p до r .

Быстрая сортировка

Парадигма «разделяй и властвуй».

Этап разделения: «опорная» книга – индекс 15



Быстрая сортировка

- 1. Разделение.** Сначала выберем одну книгу из слотов от p до r . Назовем эту книгу опорной. Представим книги на полке так, чтобы все книги с авторами, идущими в алфавитном порядке до автора опорной книги, находились слева от опорной, а книги с авторами, идущими по алфавиту после автора опорной книги, - справа от последней. После разбиения книги как слева от опорной книги, так и справа, не располагаются в каком-то конкретном порядке.
- 2. Властвование.** Осуществляется путем рекурсивной сортировки книг слева и справа от опорного элемента. То есть если при разделении опорный элемент вносится в слот q , то рекурсивно сортируются книги в слотах с p по $q-1$ и с $q+1$ по r .
- 3. Объединение.** На этом этапе мы ничего не делаем! После рекурсивной сортировки мы получаем полностью отсортированный массив. Авторы всех книг слева от опорной (в слотах с p по $q-1$) идут по алфавиту до автора опорной книги, и книги отсортированы, а авторы всех книг справа от опорной книги (с $q+1$ по r) идут по алфавиту после автора опорной книги, и все эти книги также отсортированы.

Быстрая сортировка

Процедура быстрой сортировки подразумевает вызов процедуры

Partition (A, p, r), которая разбивает подмассив $A[p..r]$ и возвращает индекс q позиции, в которую помещается опорный элемент.

Процедура *Quicksort*(A, p, r)

Вход и результат : те же, что и у процедуры *Merge-Sort*.

1. Если $p \geq r$, просто выйти из процедуры, не выполняя никаких действий.
2. В противном случае выполнить следующее.
 - A. Вызвать *Partition* (A, p, r) и установить значение q равным результату вызова.
 - B. Рекурсивно вызвать *Quicksort*($A, p, q-1$).
 - C. Рекурсивно вызвать *Quicksort*($A, q+1, r$).

Быстрая сортировка

Первоначальный вызов $QuickSort(A, l, r)$

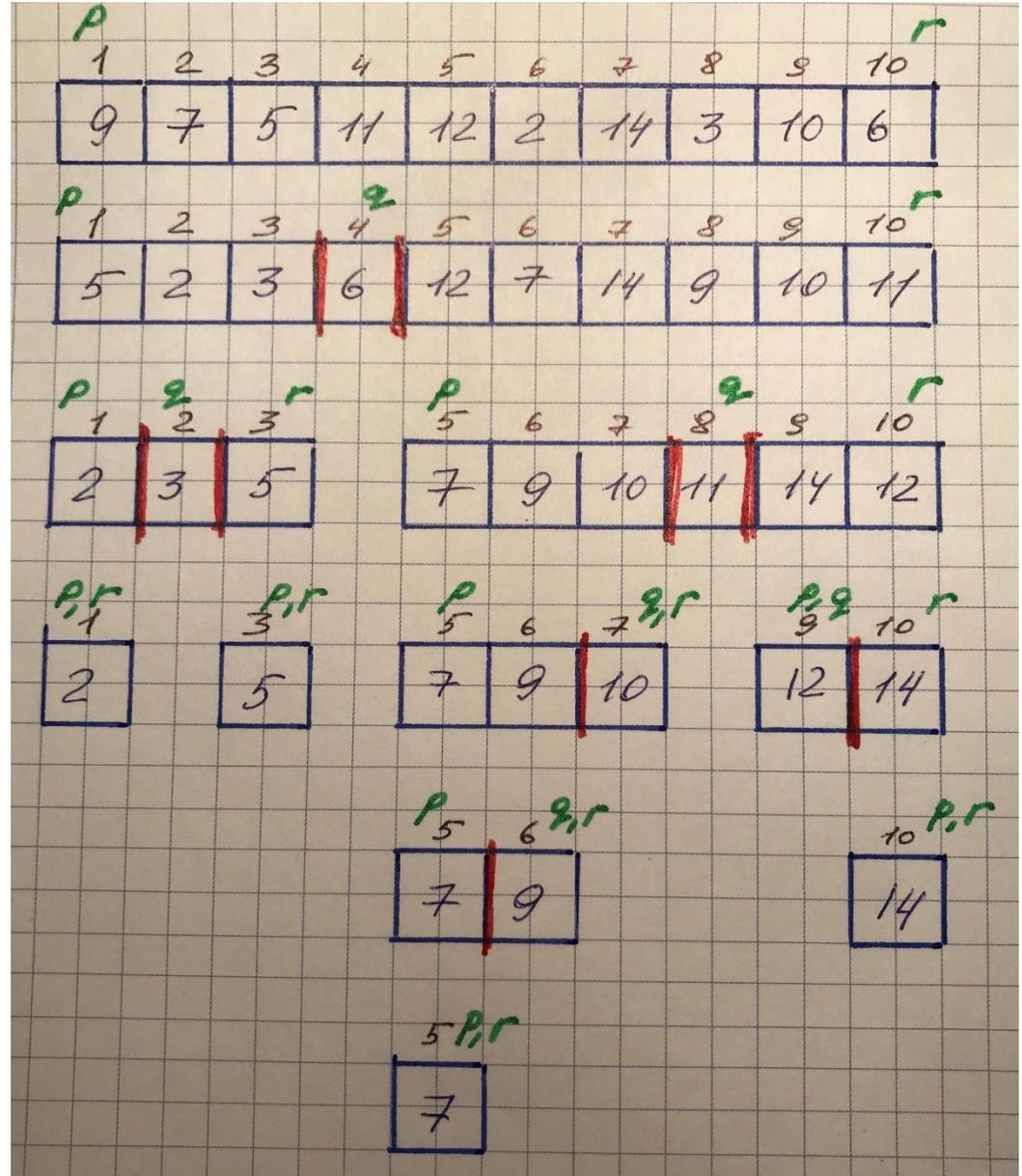
аналогичен вызову процедуры **Merge-Sort**.

Пример работы процедуры $QuickSort$ с развернутой рекурсией.

Для каждого подмассива, в котором $p \leq r$, указаны индексы p, q, r .

Самое нижнее значение в каждой позиции массива показывает, какой элемент будет находиться в этой позиции по завершении сортировки.

При чтении массива слева направо смотрим на самые нижние значения в каждой позиции, массив отсортирован.



Быстрая сортировка

Вызов процедуры Partition

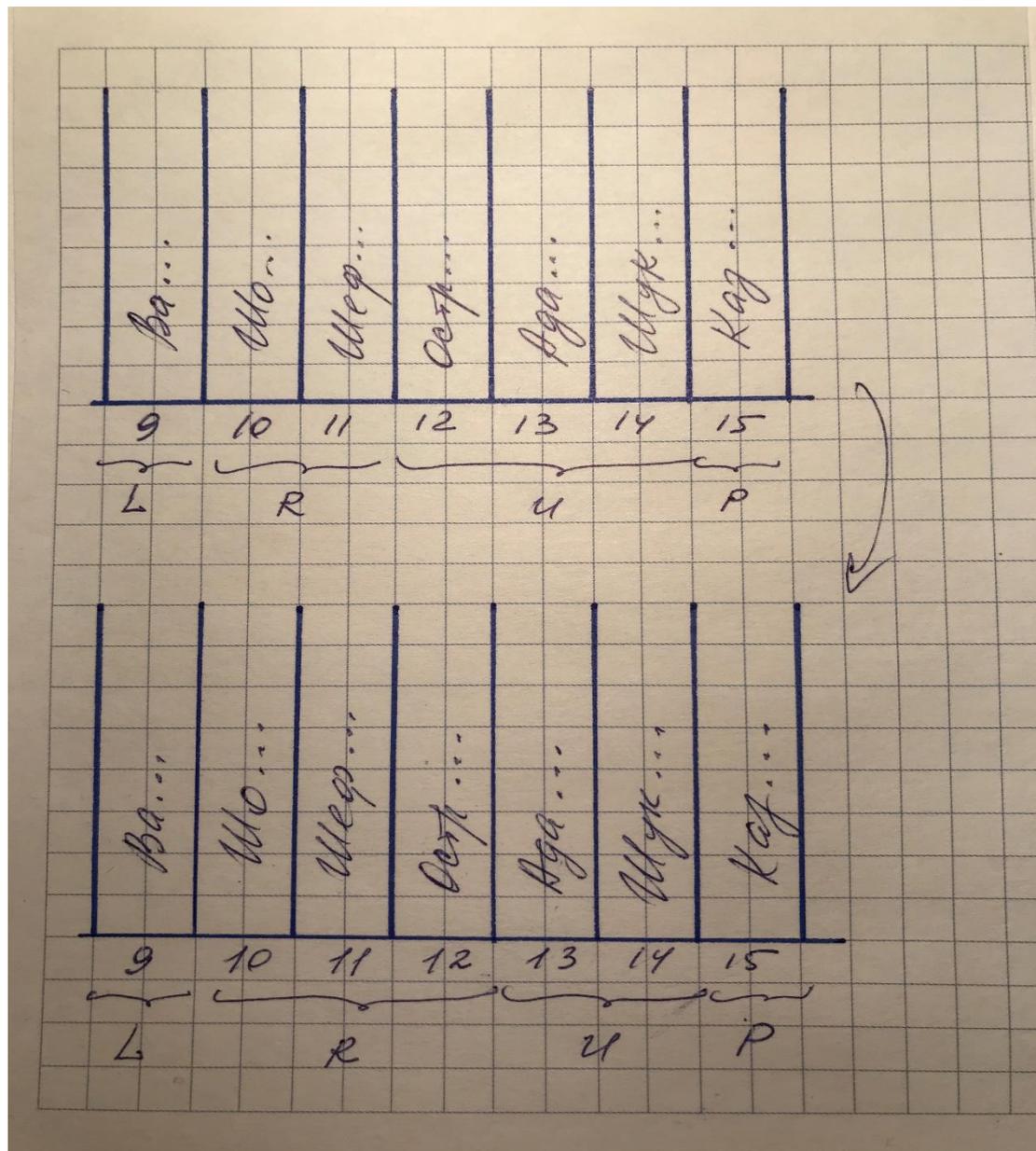
Процедура разбиения книг на полке в слотах с p по r . В качестве опорной выбираем крайнюю справа книгу - из слота r . В любой момент каждая книга может находиться в одной из четырех групп, и эти группы располагаются в слотах от p до r слева направо.

- Группа L (левая) : книги с авторами, о которых известно, что они располагаются в алфавитном порядке до автора опорной книги или написаны автором опорной книги.
- Далее идет группа R (правая): книги с авторами, о которых известно, что они располагаются в алфавитном порядке после автора опорной книги.
- Затем идет группа U (неизвестная) : книги, которые еще не рассмотрели и не знаем, как их авторы располагаются по отношению к автору опорной книги в алфавитном порядке.
- Последней идет группа P (опорная): в нее входит единственная опорная книга.

Мы проходим по книгам группы U слева направо, сравнивая каждую из них с опорной и перемещая ее либо в группу L , либо в группу R , останавливаясь по достижении опорной книги. Книга, которую мы сравниваем с опорной, - всегда крайняя слева в группе U .

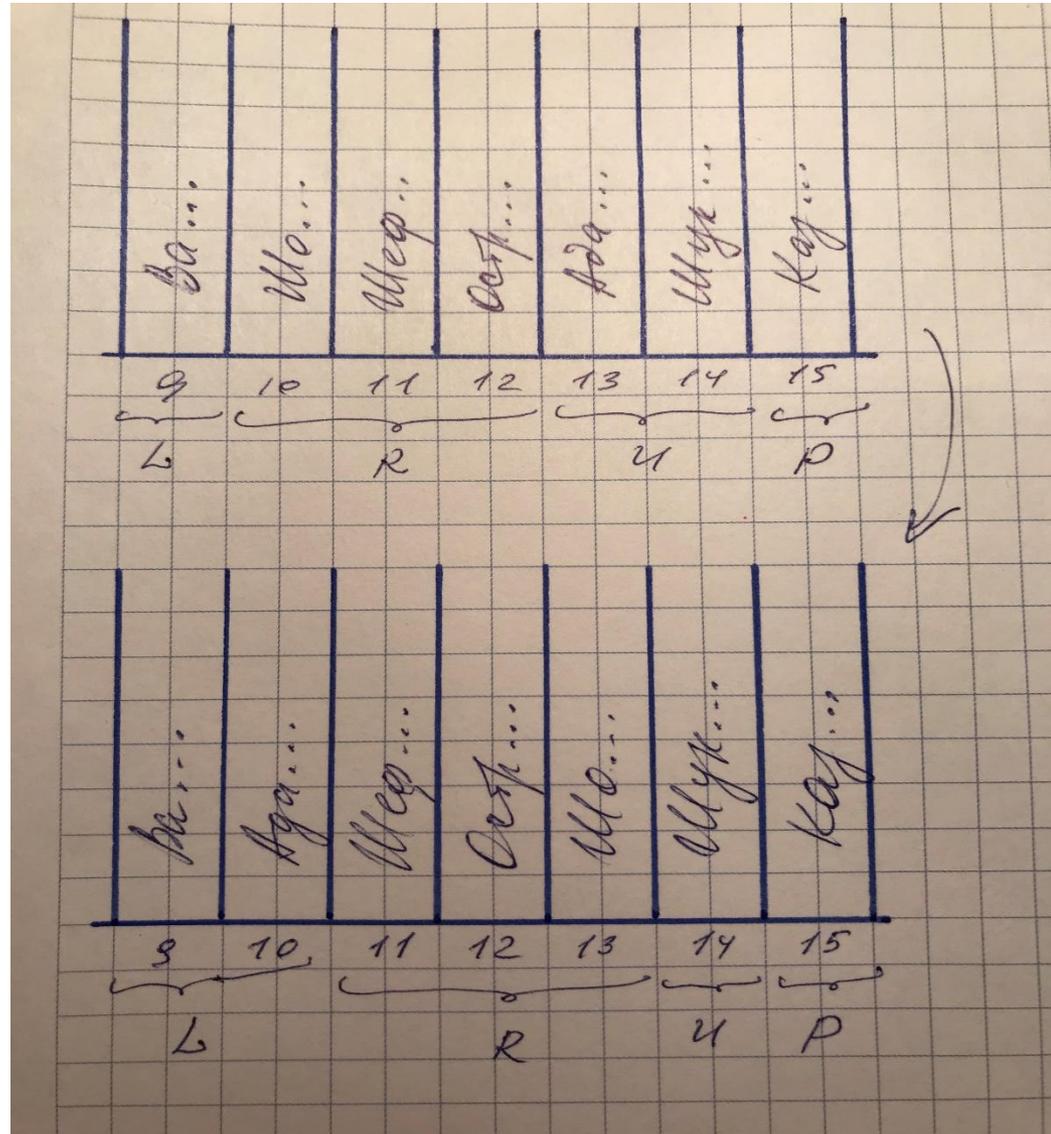
Быстрая сортировка

Вызов процедуры
Partition (A, 9,15)



Быстрая сортировка

Вызов процедуры
Partition (A, 9,15),
продолжение.



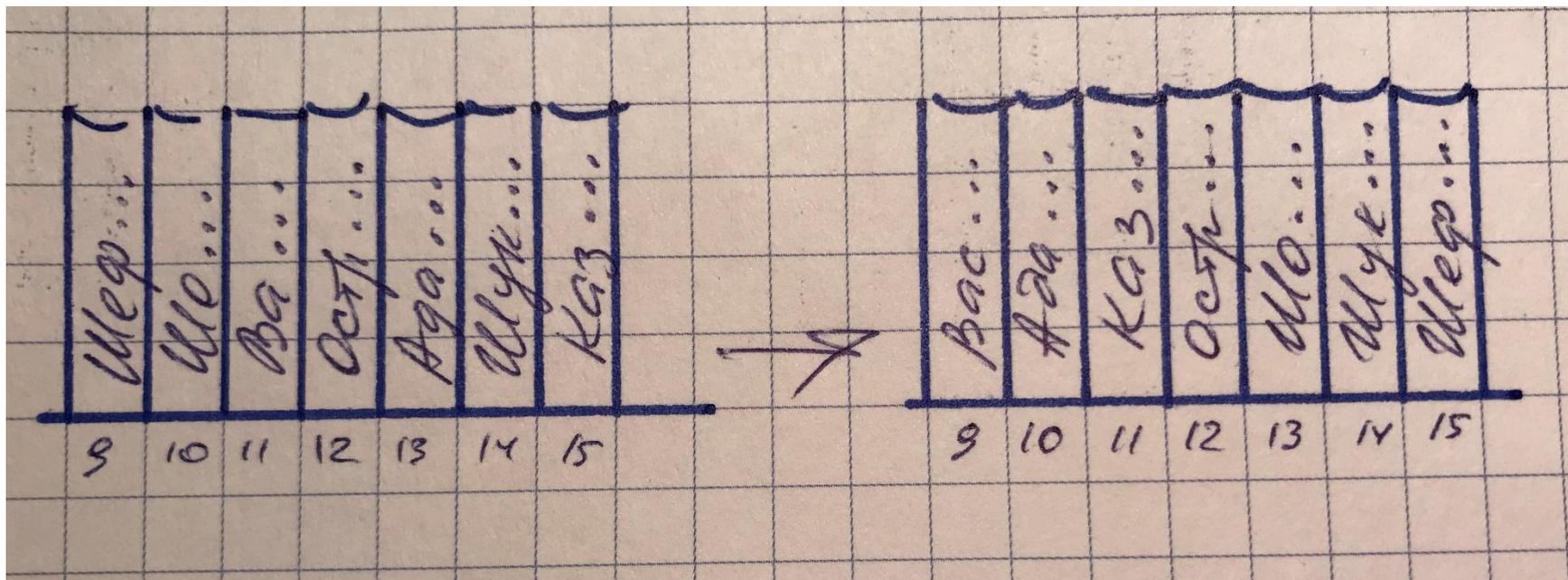
Быстрая сортировка

Вызов процедуры Partition (A, 9,15), описание

- Если автор книги находится в алфавитном порядке после автора опорной книги, то книга становится крайней справа в группе R. Поскольку до этого она была крайней слева в группе U, а за группой U непосредственно следует группа R, мы должны просто переместить разделительную линию между группами R и U на один слот вправо, без перемещения каких-либо книг.
- Если автор книги находится в алфавитном порядке до автора опорной книги или совпадает с автором опорной книги, то эта книга становится крайней справа в группе L. Мы обмениваем ее с крайней слева книгой в группе R и перемещаем разделительную линию между группами L и R и между группами R и U на один слот вправо.

Быстрая сортировка

Вызов процедуры Partition (A, 9,15), результат.



Быстрая сортировка

Чтобы преобразовать разбиение книг на полке в разбиение подмассива $A[p..r]$, мы сначала выбираем крайний справа элемент $A[r]$ в качестве опорного. Затем мы проходим через подмассив слева направо, сравнивая каждый элемент с опорным. Мы поддерживаем в подмассиве индексы q и u , которые разделяют его следующим образом:

- подмассив $A[p..q-1]$ соответствует группе L: каждый его элемент не превышает опорный;
- подмассив $A[q..u-1]$ соответствует группе R : каждый его элемент больше опорного;
- подмассив $A[u..r-1]$ соответствует группе U: нам пока неизвестно, как его элементы соотносятся с опорным;
- элемент $A[r]$ соответствует группе P: это опорный элемент.

Быстрая сортировка

Процедура *Partition*(A, p, r)

Вход: тот же, что и для процедуры *Merge-Sort*.

Результат: перестановка элементов $A[p..r]$, такая, что каждый элемент в $A[p..q-1]$ не превышает $A[q]$, а каждый элемент в $A[q+1..r]$ больше $A[q]$. Возвращает значение индекса q .

1. Установить q равным p .

2. Для $u=p$ до $r-1$:

А. Если $A[u] \leq A[r]$, обменять $A[q]$ с $A[u]$, а затем увеличить q на 1.

3. Обменять $A[q]$ с $A[r]$, а затем вернуть q .

Быстрая сортировка

Процедура **Partition**(A, p, r):

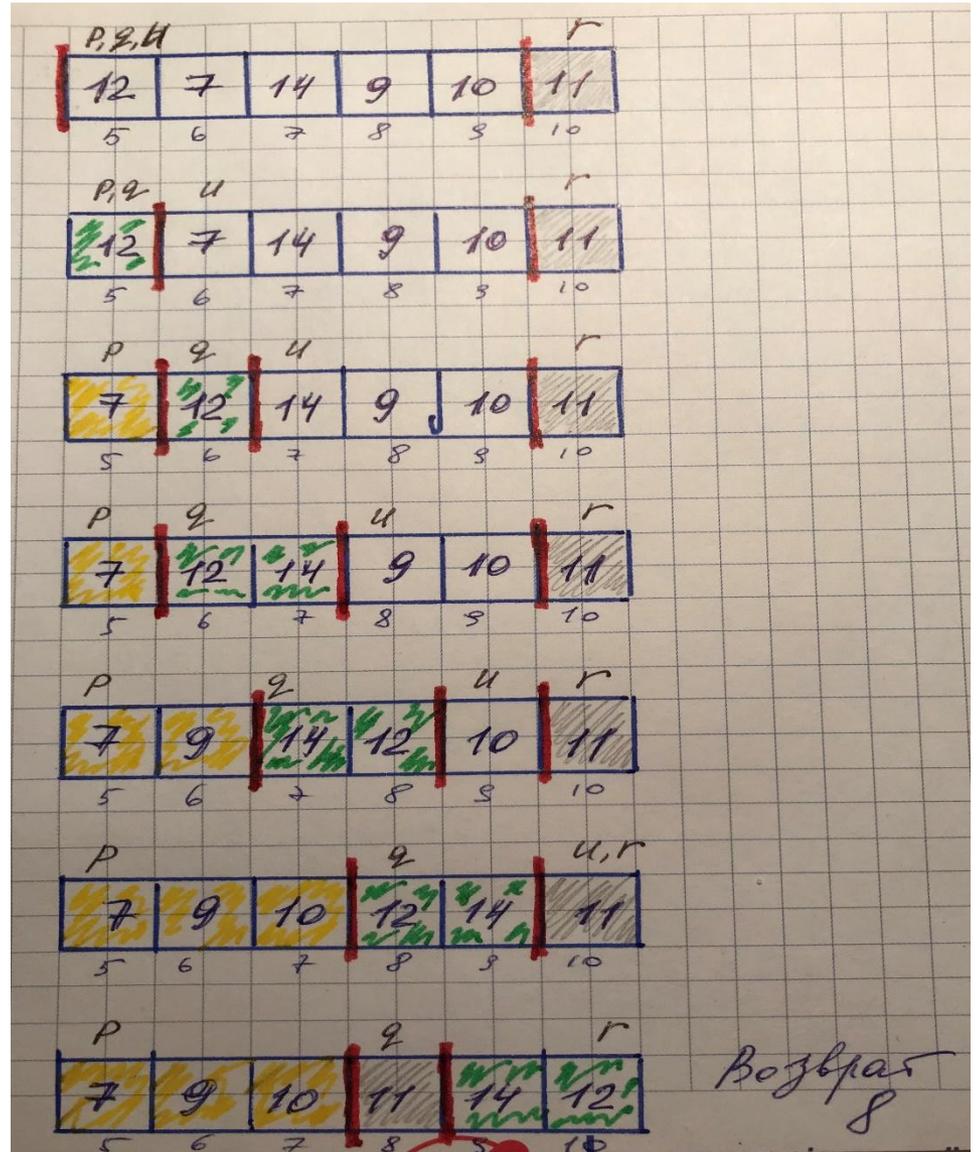
перестановка элементов $A[p..r]$, такая, что каждый элемент в $A[p..q-1]$ не превышает $A[q]$, а каждый элемент в $A[q+1..r]$ больше $A[q]$. Возвращает значение индекса q .

1. Установить q равным p .

2. Для $u=p$ до $r-1$:

Если $A[u] \leq A[r]$, обменять $A[q]$ с $A[u]$, а затем увеличить q на 1.

3. Обменять $A[q]$ с $A[r]$, а затем вернуть q .



Быстрая сортировка

На предыдущем слайде(21) показано как работает процедура Partition с подмассивом $A[5..10]$, созданным при первом разбиении в примере с быстрой сортировки на слайде 13.