ГУАП Государственный университет аэрокосмического приборостроения

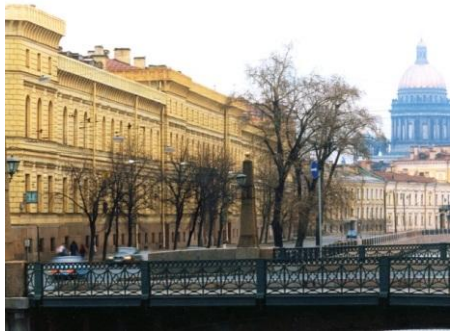
www.guap.ru

Информатика

Рождественская Ксения Николаевна

Кафедра 14

ksu.khramenkova@gmail.com



Методы сортировки

- Сортировку следует понимать как процесс перегруппировки заданного множества объектов в определенном порядке
- Сортировка применяется для облегчения поиска элементов в упорядоченном множестве. Задача сортировки одна из фундаментных в программировании

Сортировка – это упорядочивание набора однотипных данных по возрастанию или убыванию

Методы сортировки

Сортировка методом простого выбора (простой перебор)

- При сортировке массива методом выбора применяется базовый **алгоритм поиска максимального (минимального) элемента** и его номера.

Алгоритм сортировки массива методом выбора

1. Для исходного массива выбрать максимальный (минимальный) элемент
2. Поменять его местами с последним (первым) элементом (после этого самый большой(наименьший) элемент будет стоять на своем месте)
3. (для сортировки по возрастанию) Повторить п.п. 1-2 с оставшимися n-1 элементами.
   - Рассмотреть часть массива, начиная с первого элемента до предпоследнего, найти в нем максимальный элемент и поменять его местами с предпоследним (n-1)- м элементом массива, затем с оставшиеся (n-2)-мя элементами и так далее, пока не останется один элемент, уже стоящий на своем месте.
   - Аналогично для сортировки по уменьшению значения элементов.

Методы сортировки

Сортировка методом простого выбора (простой перебор)

- Для упорядочения массива потребуется (n-1) просмотров массива.
- В процессе сортировки будет увеличиваться отсортированная часть массива, а неотсортированная, соответственно, уменьшаться

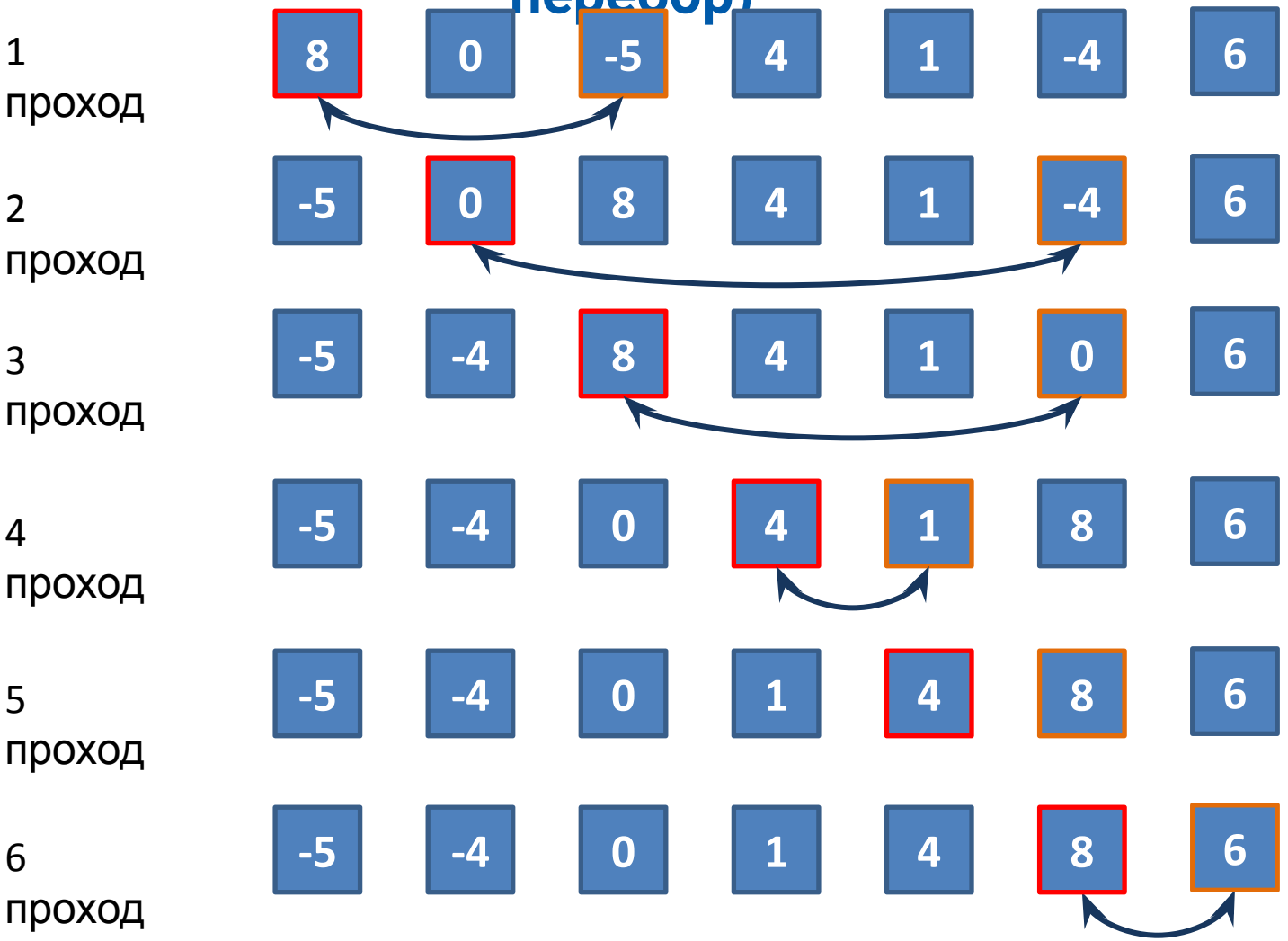
Внешний цикл алгоритма выполняется n-1 раз, а внутренний – в среднем n/2 раз. Т.е. сортировка методом простого выбора требует

$$\frac{(n^2 - n)}{2}$$ сравнений

Это алгоритм порядка n^2, из-за чего он считается слишком медленным для сортировки большого количества элементов

Методы сортировки

Сортировка методом простого выбора (простой перебор)



Методы сортировки

Сортировка методом "пузырька" (простого обмена)

- Алгоритм состоит в повторяющихся проходах по сортируемому массиву.
- За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется обмен элементов.
- Проходы по массиву повторяются до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает – массив отсортирован.
- При проходе алгоритма элемент, стоящий не на своём месте, "всплывает" до нужной позиции

Методы сортировки

Сортировка методом "пузырька" (простого обмена)

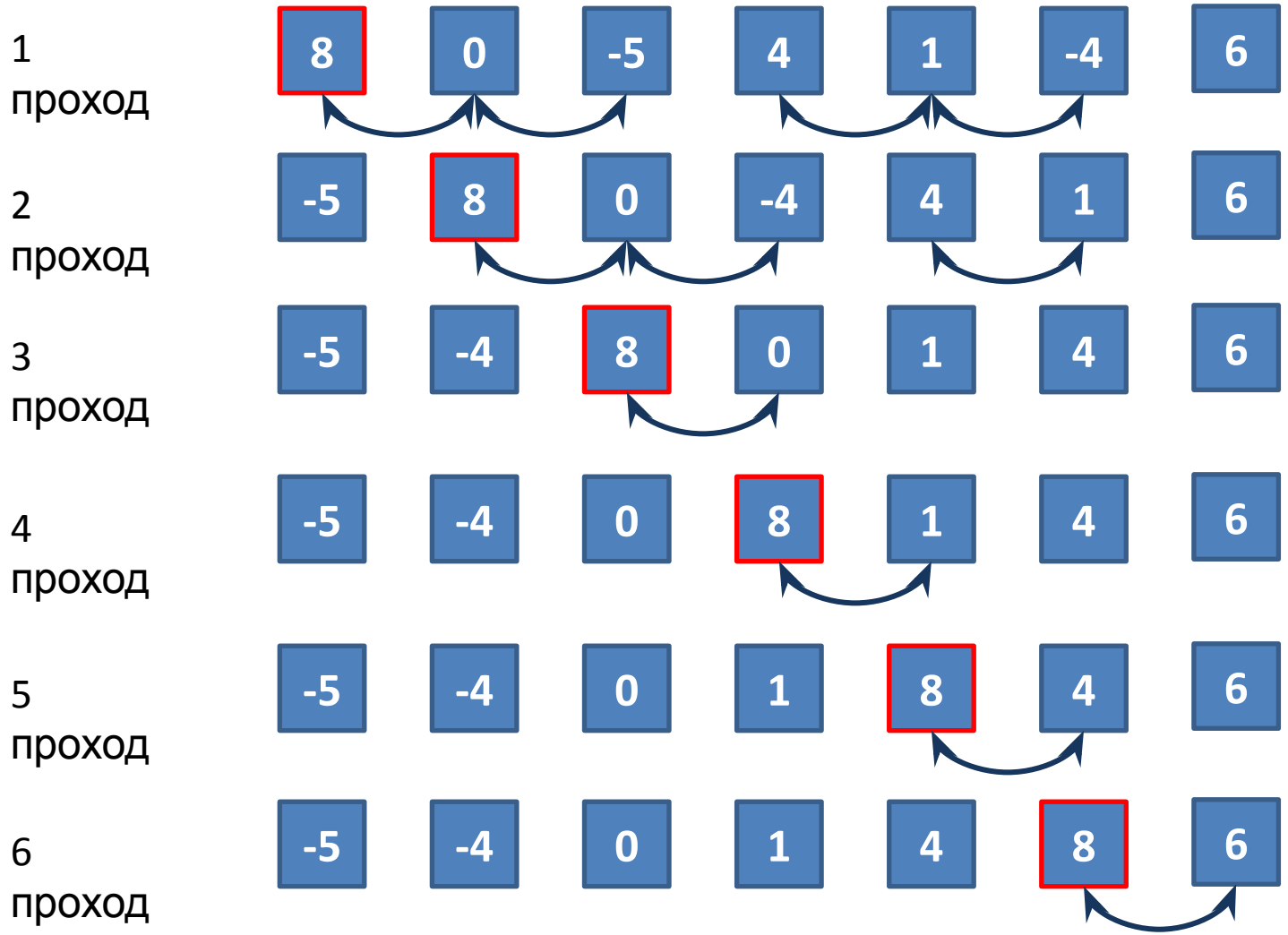
Количество сравнений всегда одно и то же, поскольку два цикла повторяются указанное количество раз. Это значит, что алгоритм всегда выполняет $\frac{(n^2-n)}{2}$ сравнений.

где n – количество сортируемых элементов

(внешний цикл выполняется n-1 раз, а внутренний выполняется в среднем n/2 раз)

Методы сортировки

Сортировка методом "пузырька" (простого обмена)



Методы сортировки

Сортировка методом "пузырька" (простого обмена)

Особенность: неупорядоченные элементы на "большом" конце массива занимают правильные положения за один проход, но неупорядоченные элементы в начале массива поднимаются на свои места очень медленно

Вместо того чтобы постоянно просматривать массив в одном направлении, в последовательных проходах можно **чередовать направления**

Это шейкер-сортировка

Время выполнения порядка N^2. Это объясняется тем, что количество сравнений не изменилось, а количество обменов уменьшилось лишь на относительно небольшую величину

Методы сортировки

Шейкер-сортировка

- Массив просматривается поочередно справа налево и слева направо.
- Просмотр массива осуществляется до тех пор, пока все элементы не встанут в порядке возрастания (убывания).
- Количество просмотров элементов массива определяется моментом упорядочивания его элементов

Методы сортировки

Сортировка пузырьком - вариант с "флагом"

Если при выполнении прохода методом пузырька не было ни одного обмена элементов массива

Массив уже отсортирован и остальные проходы не нужны

Методы сортировки

Сортировка вставками

- В начале сортировки первый элемент массива считается отсортированным, все остальные — не отсортированные.

- Начиная со второго элемента массива и заканчивая последним, алгоритм **вставляет** неотсортированный элемент массива в нужную позицию в отсортированной части массива.
  - за один шаг сортировки отсортированная часть массива увеличивается на один элемент, а неотсортированная часть массива уменьшается на один элемент

- На каждом шаге сортировки сравнивается текущий элемент со всеми элементами в отсортированной части и повторяется второй пункт

Вычислительная сложность алгоритма $O(n^2)$

Методы сортировки

Сортировка вставками

Шаг	Отсортированная часть массива	Текущий элемент
1	4	1
2	1 4	4
3	1 4 4	5
4	1 4 4 5	9
5	1 4 4 5 9	0
6	0 1 4 4 5 9	

Методы сортировки

Метод Шелла

- Метод построен на основе метода вставки с минимизацией промежуточных шагов.

- Общая схема метода состоит в следующем:

  Шаг 1. Происходит упорядочивание элементов n/2 пар $(x_i, x_{n/2+i})$ для $1 < i < n/2$

  Шаг 2. Упорядочиваются элементы в n/4 группах из четырех элементов $(x_i, x_{n/4+i}, x_{n/2+i}, x_{3n/4+i})$ для $1 < i < n/4$

  Шаг 3. Упорядочиваются элементы уже в n/4 группах из восьми элементов и т.д.

- На последнем шаге упорядочиваются элементы сразу во всем массиве x1,x2,...,xn.

- На каждом шаге для упорядочивания элементов в группах используется метод сортировки вставками

Методы сортировки

Метод Шелла

Исходный массив чисел

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
5	3	8	0	7	4	9	1	6	2

Шаг 1. 10/2 = 5. Числа расположены на расстоянии 5 друг от друга.
Список пар следующий: (5,4), (3,9), (8,1), (0,6), (7,2).
Отсортируем внутри пары по возрастанию и расставим в исходном массиве

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
4	3	1	0	2	5	9	8	6	7

Методы сортировки

Метод Шелла

Шаг 2. 5/2=2. Числа расположены на расстоянии 2 друг от друга. Отсортируем внутри пары по возрастанию и расставим в исходном массиве. Выполняем сортировку последовательно.

Пара (4,1):

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	3	4	0	2	5	9	8	6	7

Пара

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	0	4	3	2	5	9	8	6	7

Пара

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	0	2	3	4	5	9	8	6	7

Методы сортировки

Метод Шелла

Пара (3,5):

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	0	2	3	4	5	9	8	6	7

Пара

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	0	2	3	4	5	9	8	6	7

Пара

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	0	2	3	4	5	9	8	6	7

Методы сортировки

Метод Шелла

Пара (9,6):

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	0	2	3	4	5	6	8	9	7

Пара (8,7):

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	0	2	3	4	5	6	7	9	8

Методы сортировки

Метод Шелла

Шаг 3. 2/2 = 1. Числа расположены на расстоянии 1 друг от друга. Отсортируем внутри пары по возрастанию и расставим в исходном массиве.

Выполняем сортировку последовательно как на предыдущем шаге.

Резул

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	1	2	3	4	5	6	7	8	9

При удачном раскладе этот способ сортирует за O(n).

Средняя временная сложность зависит от того, какую последовательность брать для циклических итераций. Первоначально автор сортировки, Дональд Шелл, предложил ряд[n/4], [n/2], [n/8], ..., который давал скорость $O(n^2)$.

Методы сортировки

Сортировка слиянием

- **Слияние** означает объединение двух (или более) последовательностей в одну упорядоченную последовательность при помощи циклического выбора элементов, доступных в данный момент
- Процедура слияния предполагает объединение двух предварительно упорядоченных подпоследовательностей размерности n/2 в единую последовательность размерности n.

Достоинством сортировки слиянием является то, что он удобен для структур с последовательным доступом к элементам

Недостаток : метод требует дополнительной памяти по объему равной объему сортируемого файла.

Методы сортировки

Сортировка слиянием

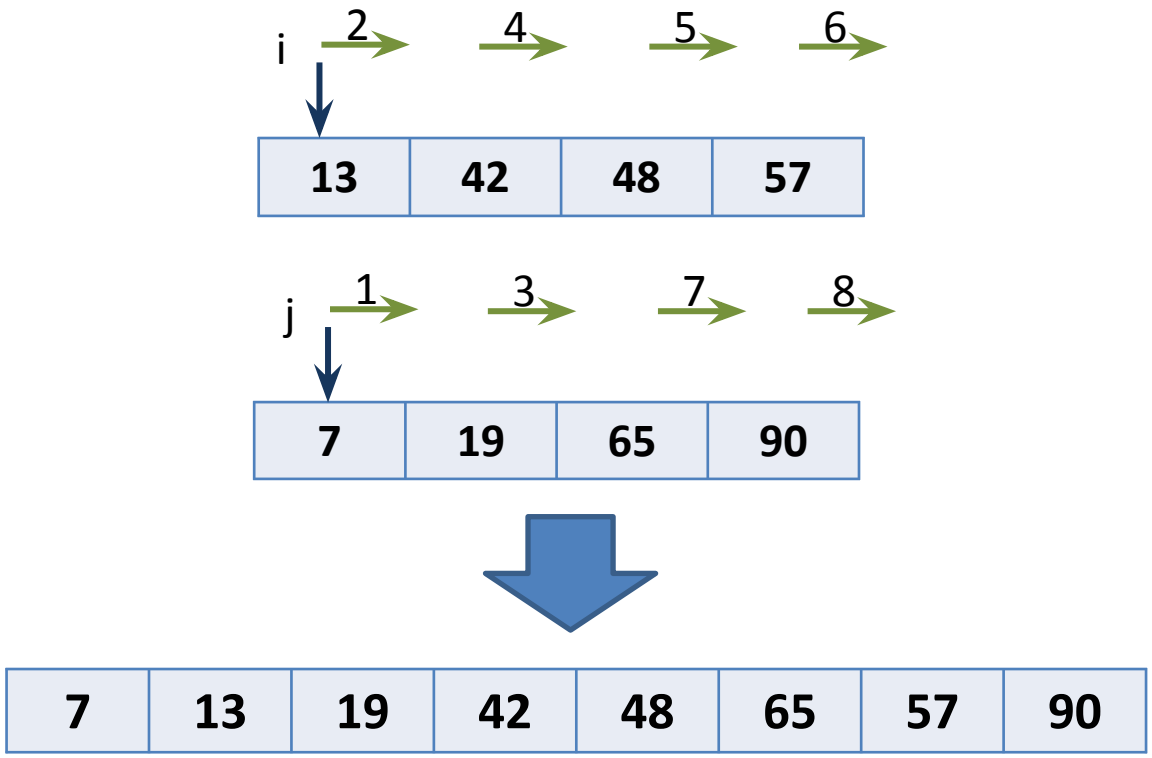
1. Элементы предварительно упорядоченных последовательностей сравниваются между собой, и из них выбирается наименьший
2. Соответствующий указатель перемещается на следующий элемент

пока не достигнут конец одной из подпоследовательностей

Оставшиеся элементы другой подпоследовательности при этом передаются в результирующую последовательность в неизменном виде

Методы сортировки

Сортировка слиянием



Методы сортировки

Сортировка слиянием

Алгоритм двухпутевого слияния (1/3)

- Исходная последовательность разбивается на две подпоследовательности

43	50	15	42	95	19	7	65

43	50	15	42

95	19	7	65

- Эти две подпоследовательности объединяются в одну, содержащую упорядоченные пары

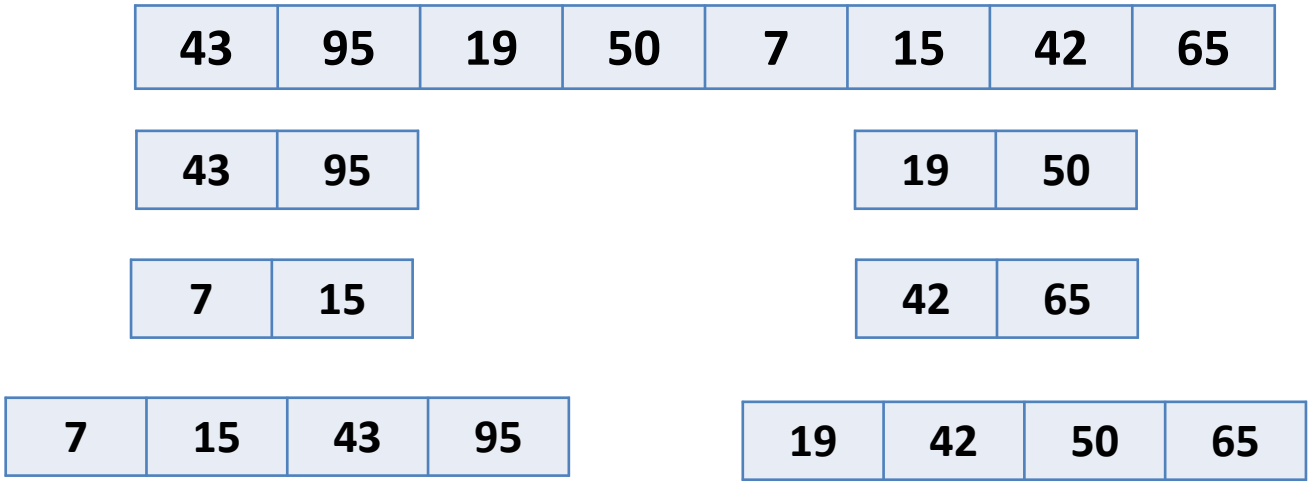
43	95	19	50	7	15	42	65

Методы сортировки

Сортировка слиянием

Алгоритм двухпутевого слияния (2/3)

- Полученная последовательность снова разбивается на две, и пары объединяются в упорядоченные четверки



Методы сортировки

Сортировка слиянием

Алгоритм двухпутевого слияния (3/3)

- Полученная последовательность снова разбивается на две и собирается в упорядоченную восьмерку

7	15	43	95

19	42	50	65

7	15	19	42	43	50	65	95

Операция повторяется до тех пор, пока полученная упорядоченная последовательность не будет иметь такой же размер, как у сортируемой

Методы сортировки

Сортировка слиянием

Метод нисходящего слияния (1/2)

- Исходная последовательность рекурсивно разбивается на половины, пока не получим подпоследовательности по 1 элементу.
- Из полученных подпоследовательностей формируем упорядоченные пары методом слияния, затем - упорядоченные четверки и т.д.

43	50	15	42	95	19	7	65

❖ Разбиваем последовательность на 2 половины (рекурсивно, пока не получим пары)

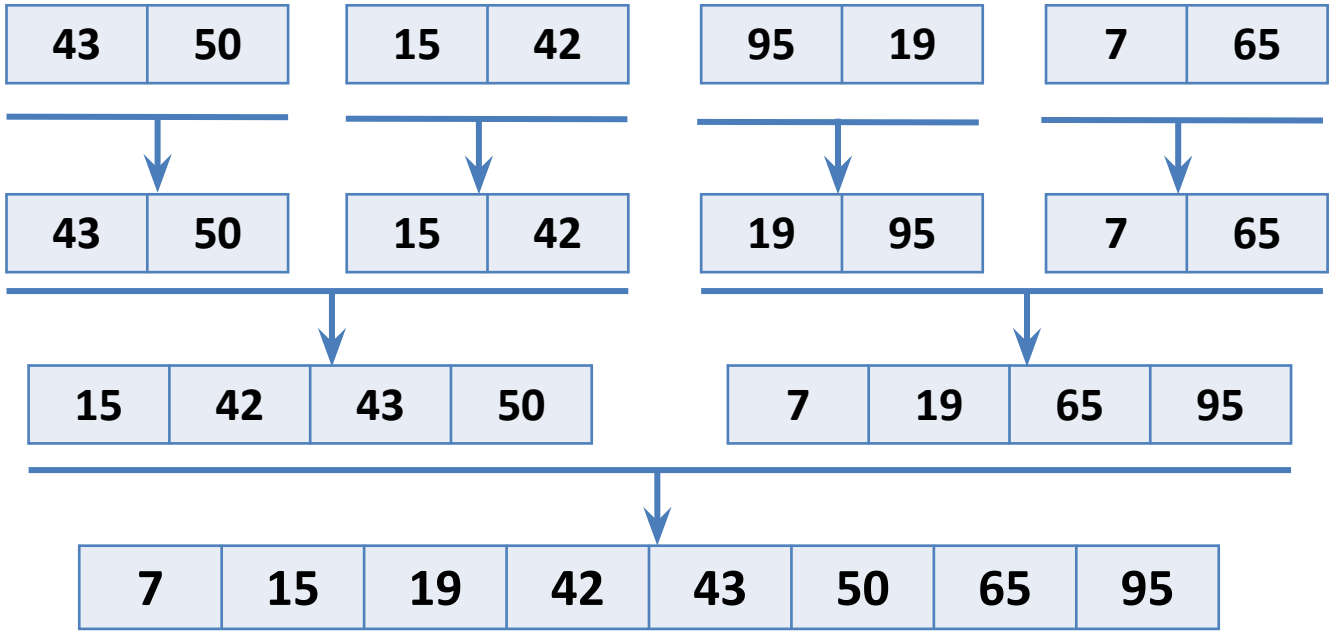
43	50		15	42		95	19		7	65

Методы сортировки

Сортировка слиянием

Метод нисходящего слияния (2/2)

- Каждую подпоследовательность упорядочиваем методом слияния и получаем готовую последовательность

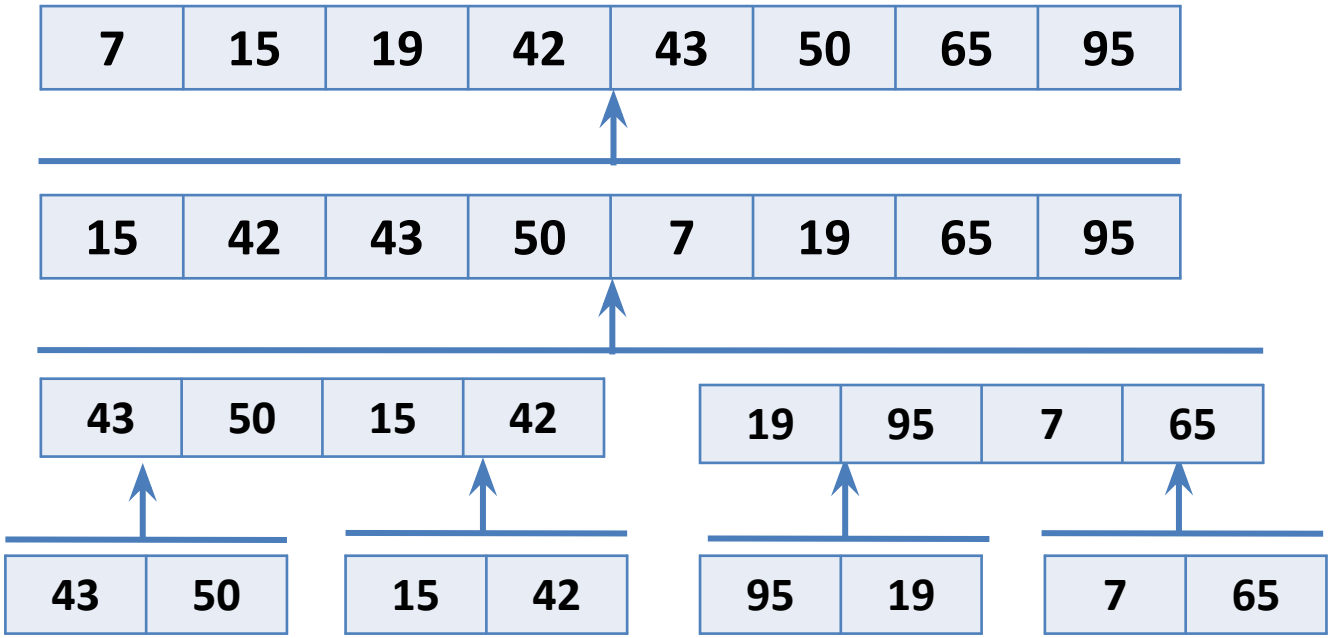


Методы сортировки

Сортировка слиянием

Метод восходящего слияния

- Исходная последовательность представляется как последовательный набор отдельных элементов



Методы сортировки

Рекурсия

- Решение той или иной задачи может быть выражено как комбинация или модификация решений той же задачи
- **Рекурсивный алгоритм** – решение задачи в ходе выполнения обращающееся само к себе.
- Любой рекурсивный алгоритм может быть описан нерекурсивно, но не наоборот
- Рекурсивный алгоритм обязательно должен содержать **две части**:
  - **Шаг рекурсии**. Указание, каким образом производится рекурсивный вызов.
  - **База рекурсии**. Условие выхода из рекурсии

Отсутствие базы рекурсии приводит к зацикливанию алгоритма

Методы сортировки

Рекурсия

Пример

Задача о числе разбиений

Найти число способов, каким можно разбить целое положительное число N на сумму целых положительных чисел

$$N = n_1+n_2+\ldots+n_m,\ n_i>0$$

Решение

- Не будем различать разбиения, отличающиеся только перестановкой слагаемых

  Например, 4+2 и 2+4 числа 6

Методы сортировки

Рекурсия

Пример

Все возможные разбиения числа 6:

6=6,
6 = 5+1,
6 = 4+2,
6 = 4+1+1,
6 = 3+3,
6 = 3+2+1,
6 = 3+1+1+1,
6 = 2+2+2,
6 = 2+2+1+1,
6 = 2+1+1+1+1,
6 = 1+1+1+1+1+1

Методы сортировки

Рекурсия

Пример

P(N) - количество разбиений числа N

✔ число разбиений можно сводить к числу разбиений слагаемых, входящих в уже учтенные суммы

 из разбиения (5+1) можно получить другие разбиения числа 6, находя их из разбиения числа 5

В данной задаче можно использовать рекурсивные вызовы

Методы сортировки

Рекурсия

Пример

- Пусть имеем какое-то разбиение и в нем максимальное слагаемое M, т.е. $n_i <= M$
- **Q(N,M)** - количество разбиений числа N слагаемыми, не превышающими M

- Если в разбиении **есть слагаемое**, точно равное M, можно вычесть его из N, и далее искать разбиение числа (N-M)

- Если в разбиении **нет слагаемых** больших или равных M, можно считать, что на самом деле ищем Q(N,M-1), тогда шаг рекурсии:

$$Q(N,M) = Q(N,M-1)+Q(N-M,M)$$

Методы сортировки

Рекурсия

Пример

- Тогда P(N) может быть выражено как **P(N)=Q(N,N)**
- На первом шаге всегда можно учесть тривиальное разбиение N=N и получить
  Q(N,N) = Q(N,N-1)+1
- Из Q(N,M) = Q(N,M-1)+Q(N-M,M) следует, что **Q(N,M)=Q(N,N)**, если N<M
- Добавив базу рекурсии, получим следующий набор выражений:

P(N) = Q(N,N)

Q(N,M) = Q(N,M-1)+Q(N-M,M), M<N

Q(N,M) = Q(N,N-1)+1, M=N

Q(N,M) = Q(N,N)

Q(1,M) = 1

Q(N,1) = 1

Методы сортировки

Рекурсия

Пример

Применим полученные рекурсивные выражения для вычисления P(6)

P(6) = Q(6,6) =

1+Q(6,5)*(в силу P(N)=Q(N,N) и Q(N,M)=Q(N,N-1)+1)* =

1+Q(6,4)+Q(1,5)*(в силу Q(N,M) = Q(N,M-1)+Q(N-M,M))* =

2+Q(6,3)+Q(2,4)*(в силу Q(N,M) = Q(N,M-1)+Q(N-M,M) и Q(1,M))* =

2+Q(6,2)+Q(3,3)+Q(2,2) … =

4+Q(6,1)+Q(4,2)+Q(3,2)+Q(2,1) =

6+Q(4,1)+Q(2,2)+Q(3,1)+Q(1,2) =

10+Q(2,1) = **11**

Методы сортировки

Метод Хоара для упорядочивания массива

- **Быстрая сортировка** представляет собой усовершенствованный метод сортировки, основанный на принципе обмена
- Для достижения наибольшей эффективности желательно производить обмен элементов на больших расстояниях

- В массиве выбирается некоторый элемент, называемый разрешающим
- Он помещается в то место массива, где ему полагается быть после упорядочивания всех элементов

В процессе отыскания подходящего места для разрешающего элемента производятся перестановки элементов так, что **слева** от них находятся **элементы, меньшие разрешающего**, и **справа** — **больше**

(предполагается, что массив сортируется по возрастанию)

Методы сортировки

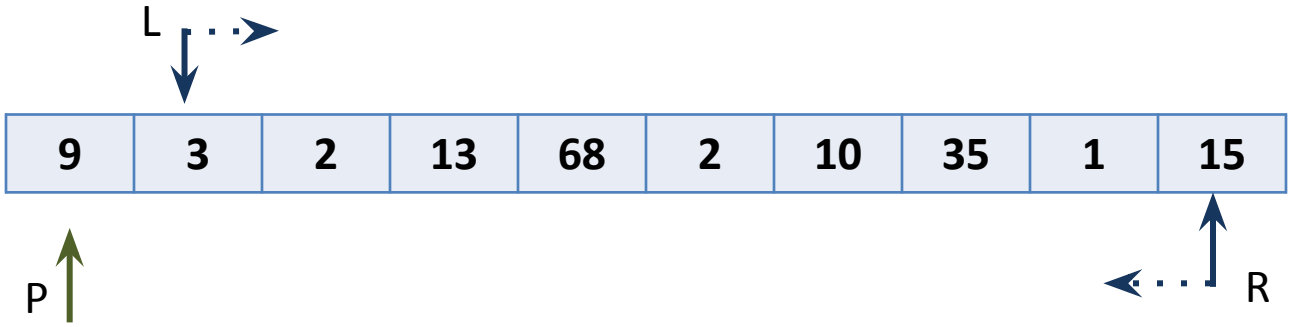
Метод Хоара для упорядочивания массива

- Тем самым массив разбивается на две части
  - не отсортированные элементы слева от разрешающего элемента
  - не отсортированные элементы справа от разрешающего элемента
- Чтобы отсортировать эти два меньших подмассива, алгоритм рекурсивно вызывает сам себя
- Ключевым элементом быстрой сортировки является **алгоритм переупорядочения**

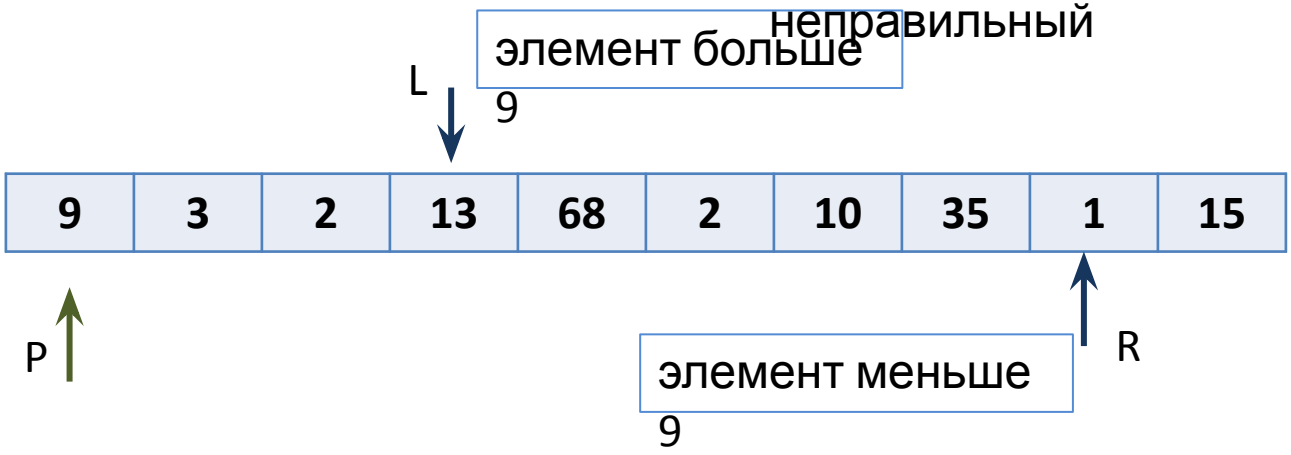
Методы сортировки

Метод Хоара для упорядочивания массива

Пример



Движение указателей останавливается, когда встречаются элементы, порядок расположения которых относительно разрешающего элемента неправильный

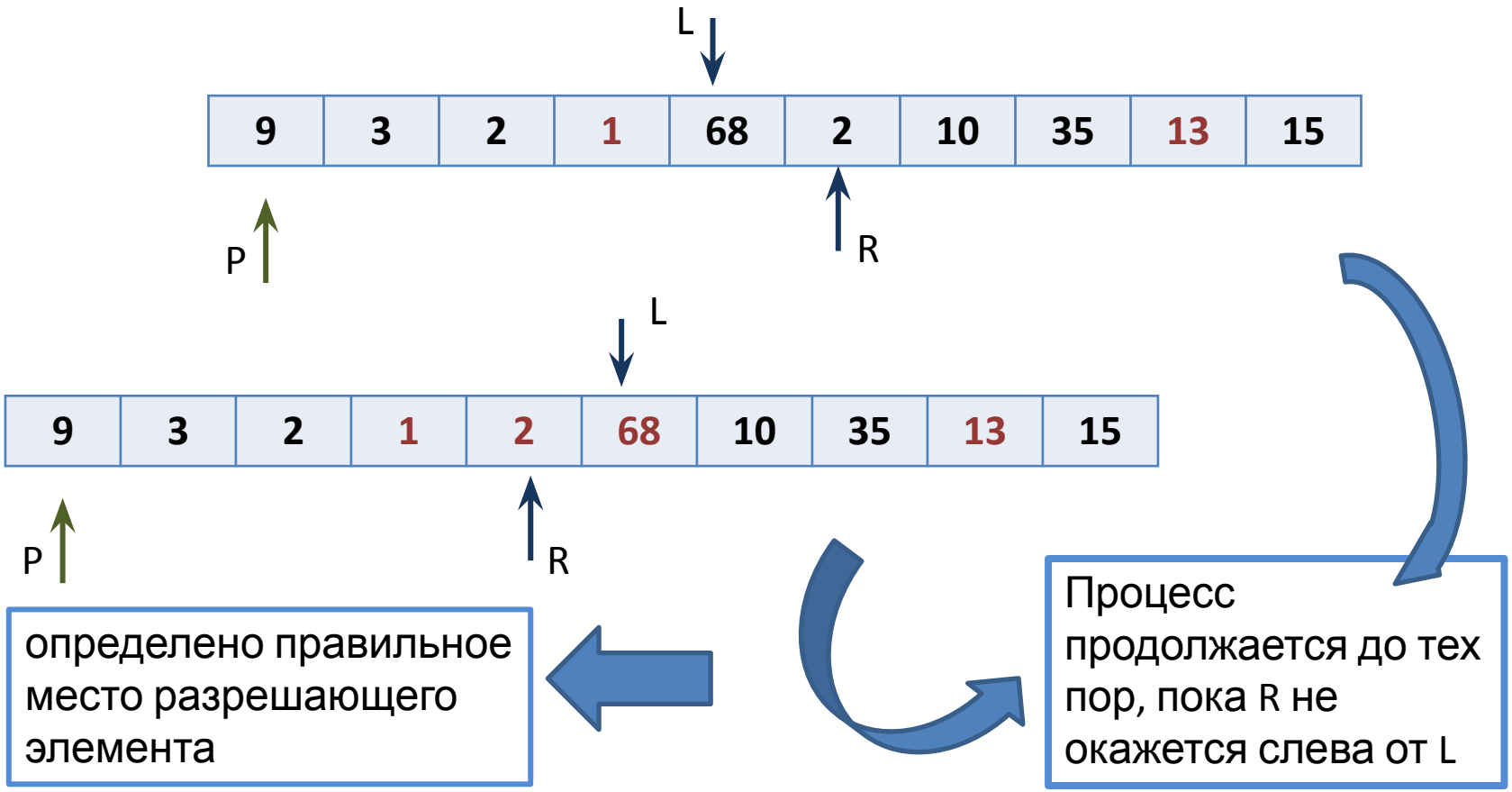


Найденные элементы меняются местами и движение указателей возобновляется

Методы сортировки

Метод Хоара для упорядочивания массива

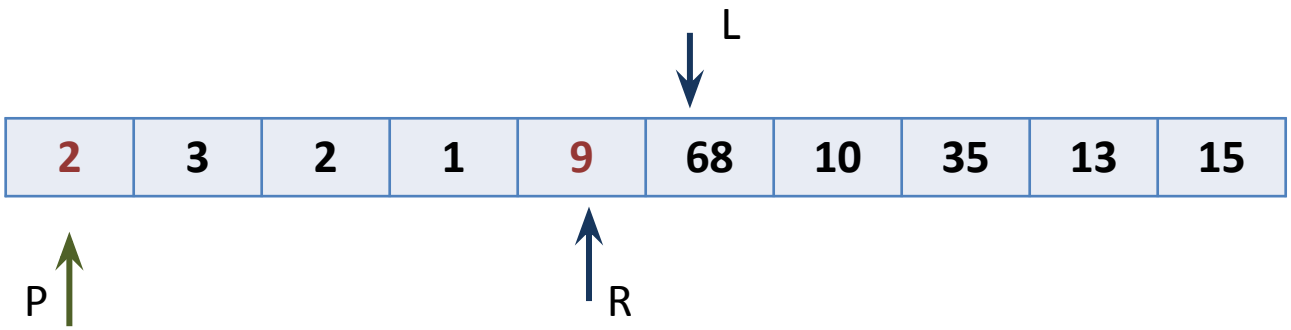
Пример



Методы сортировки

Метод Хоара для упорядочивания массива

Пример



Перестановка разрешающего элемента с элементом, на который указывает R

- Разрешающий элемент находится в нужном месте: элементы слева от него имеют меньшие значения; справа — большие

Алгоритм рекурсивно вызывается для сортировки подмассивов слева от разрешающего и справа от него