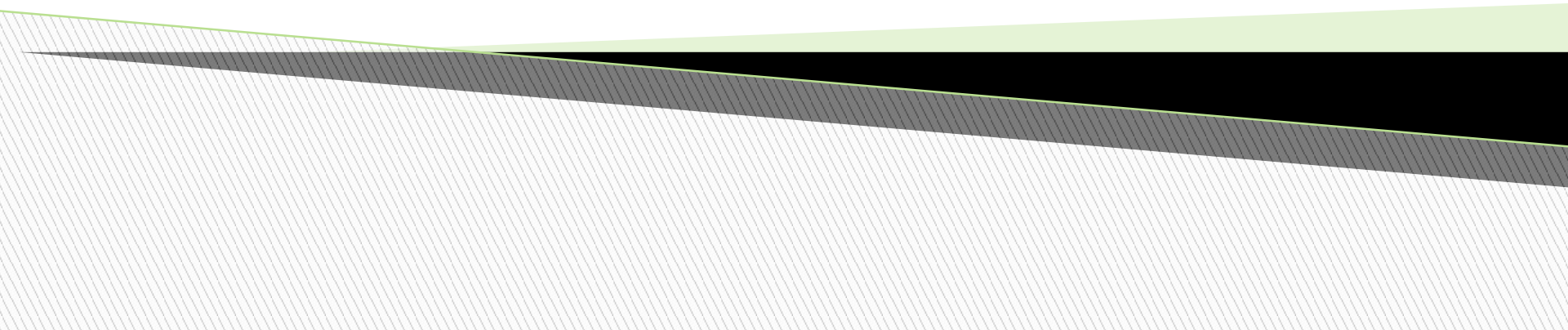
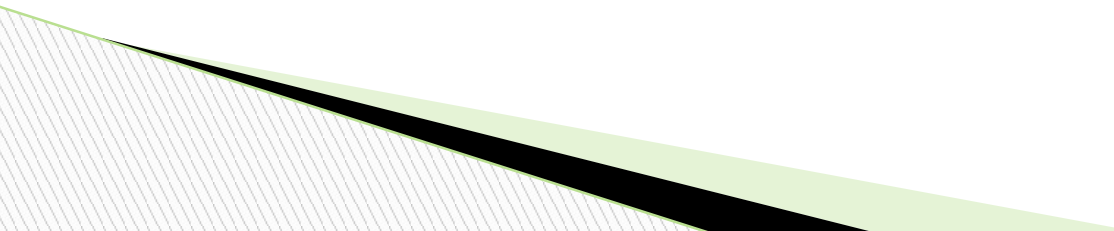


ОДНОМЕРНЫЕ МАССИВЫ: ЗАДАЧИ СОРТИРОВОК ЭЛЕМЕНТОВ МАССИВА



В общем случае сортировку следует понимать как процесс перегруппировки заданного множества объектов в определенном порядке. Часто при сортировке больших объемов данных нецелесообразно переставлять сами элементы, поэтому для решения задачи выполняется упорядочивание элементов по индексам. То есть индексы элементов выстраивают в такой последовательности, что соответствующие им значения элементов оказываются отсортированными по условию задачи

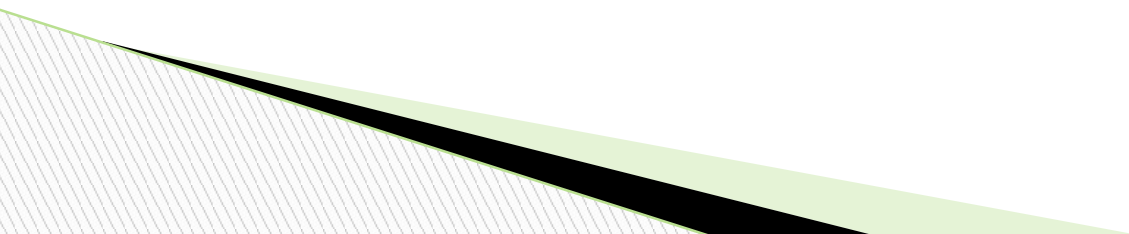


- Сортировка применяется для облегчения поиска элементов в упорядоченном множестве. Задача сортировки одна из фундаментальных в программировании.

Сортировка – это упорядочивание набора однотипных данных по возрастанию или убыванию.

- Чаще всего при сортировке данных лишь часть их используется в качестве ключа сортировки. **Ключ сортировки** – это часть данных, определяющая порядок элементов. Таким образом, ключ участвует в сравнениях, но при обмене элементов происходит перемещение всей структуры данных. Например, в списке почтовой рассылки в качестве ключа может использоваться почтовый индекс, но сортируется весь адрес. При решении задач сортировок массивов ключ и данные совпадают.

АЛГОРИТМЫ СОРТИРОВКИ



▣ **Скорость работы** алгоритма сортировки. Она непосредственно связана с количеством сравнений и количеством обменов, происходящих во время сортировки, причем обмены занимают больше времени. Сравнение происходит тогда, когда один элемент массива сравнивается с другим; обмен происходит тогда, когда два элемента меняются местами. Время работы одних алгоритмов сортировки растет экспоненциально, а время работы других логарифмически зависит от количества элементов.

Время работы в лучшем и худшем случаях.

Оно имеет значение при анализе выполнения алгоритма, если одна из крайних ситуаций будет встречаться довольно часто. Алгоритм сортировки зачастую имеет хорошее среднее время выполнения, но в худшем случае он работает очень медленно.

Поведение алгоритма сортировки.

Поведение алгоритма сортировки называется естественным, если время сортировки минимально для уже упорядоченного списка элементов, увеличивается по мере возрастания степени неупорядоченности списка и максимально, когда элементы списка расположены в обратном порядке. Объем работы алгоритма оценивается количеством производимых сравнений и обменов.

Различные сортировки массивов отличаются по быстродействию. Существуют простые методы сортировок, которые требуют порядка n^2 сравнений, где n – количество элементов массива и быстрые сортировки, которые требуют порядка $n \cdot \ln(n)$ сравнений. Простые методы удобны для объяснения принципов сортировок, т.к. имеют простые и короткие алгоритмы. Усложненные методы требуют меньшего числа операций, но сами операции более сложные, поэтому для небольших массивов простые методы более эффективны.

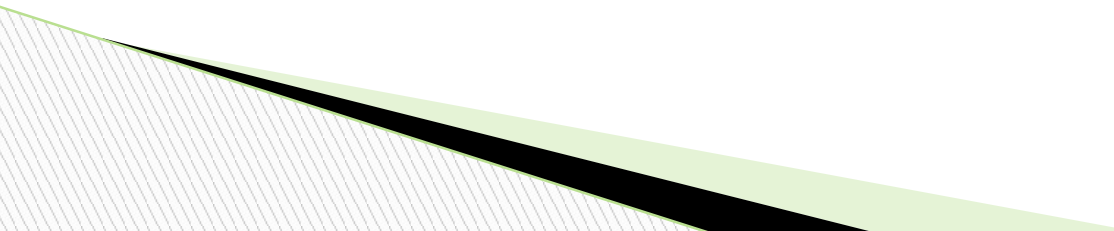
Простые методы сортировки можно разделить на три основные категории:

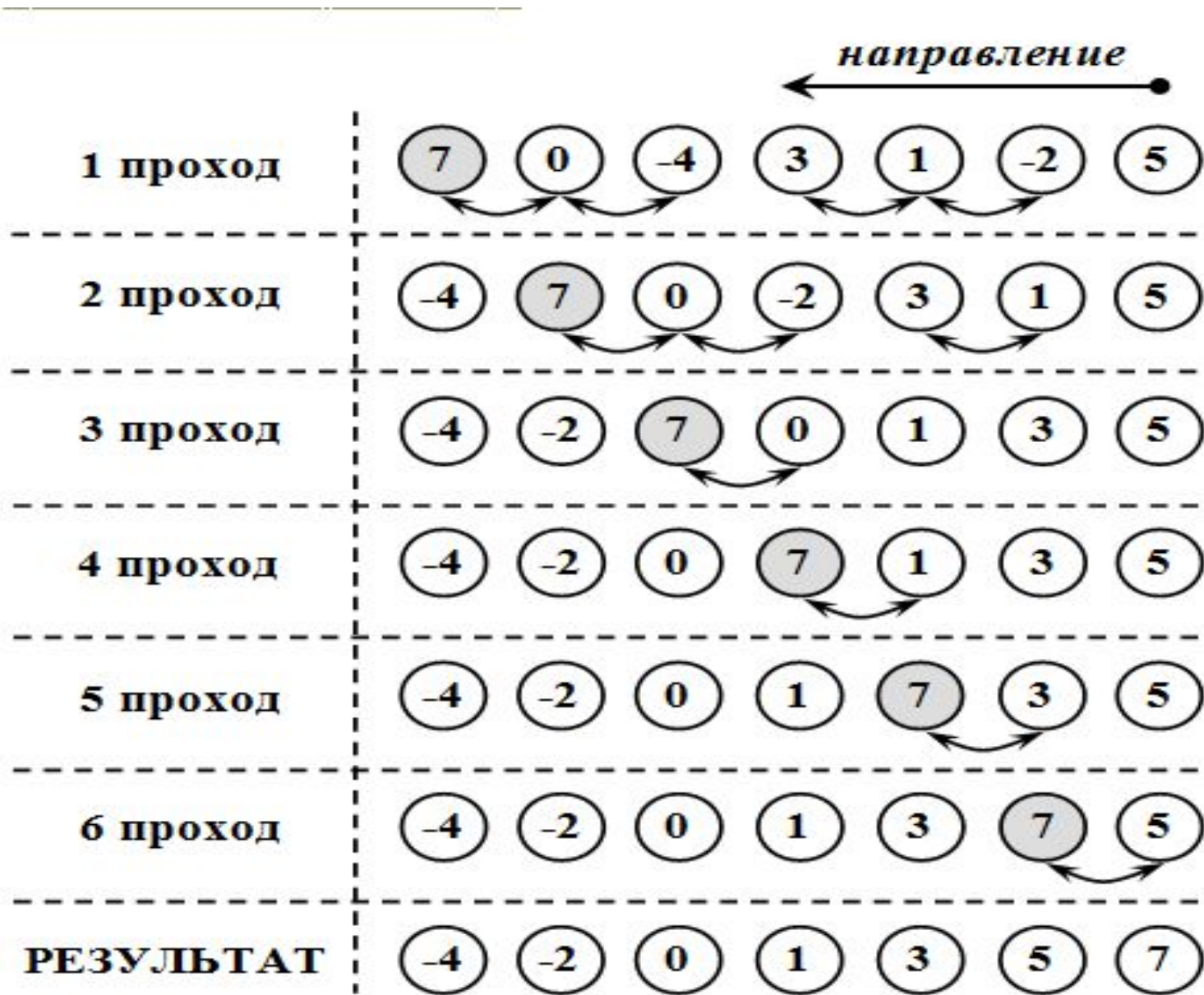
- сортировка методом "пузырька" (простого обмена);
- сортировка методом простого выбора (простой перебор);
- сортировка методом простого включения (сдвиг-вставка, вставками, вставка и сдвиг).

Сортировка методом "пузырька" (простого обмена)

Алгоритм попарного сравнения элементов массива в литературе часто называют "методом пузырька", проводя аналогию с пузырьком, поднимающимся со дна бокала с газированной водой. По мере всплывания пузырек сталкивается с другими пузырьками и, сливаясь с ними, увеличивается в объеме. Чтобы аналогия стала очевидной, нужно считать, что элементы массива расположены вертикально друг над другом, и их нужно так упорядочить, чтобы они увеличивались сверху вниз.

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.





В пузырьковой сортировке количество сравнений всегда одно и то же, поскольку два цикла for повторяются указанное количество раз независимо от того, был список изначально упорядочен или нет. Это значит, что алгоритм пузырьковой сортировки всегда выполняет сравнений, где n – количество сортируемых элементов. Данная формула выведена на том основании, что внешний цикл выполняется $n-1$ раз, а внутренний выполняется в среднем $n/2$ раз

Пузырьковая сортировка имеет такую 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. Поэтому, вместо того чтобы постоянно просматривать массив в одном направлении, в последовательных проходах можно чередовать направления. Таким образом, элементы, сильно удаленные от своих положений, быстро станут на свои места. Данная версия пузырьковой сортировки носит название шейкер-сортировки (shaker sort сортировка перемешиванием, сортировка взбалтыванием, сортировка встряхиванием), поскольку действия, производимые ею с массивом, напоминают взбалтывание или встряхивание. Ниже показана реализация шейкер-сортировки.

```
Public Sub sort()  
Dim n, i, buf As Integer  
Dim t As Boolean  
Dim A() As Integer  
n = InputBox("Введите количество элементов", "Ввод данных")  
ReDim A(1 To n)  
Randomize  
Debug.Print "Исходный массив:"  
For i = 1 To n  
    A(i) = 19 * Rnd  
    Debug.Print A(i)  
Next i  
Debug.Print  
t = True  
Do While t  
    t = False  
    For i = 1 To n - 1  
        If A(i) > A(i + 1) Then  
            buf = A(i)  
            A(i) = A(i + 1)  
            A(i + 1) = buf  
            t = True  
        End If  
    Next i  
Loop  
Debug.Print "Полученный массив:"  
For i = 1 To n  
    Debug.Print A(i)  
Next i  
Debug.Print  
End Sub
```