1. ЭЛЕМЕНТАРНЫЕ МЕТОДЫ СОРТИРОВКИ МАССИВОВ

Элементарные методы сортировки массивов – это простые и базовые алгоритмы сортировки, которые обычно используются на небольших массивах или в качестве основы для более эффективных алгоритмов. Вот некоторые из таких методов:

* 1. Сортировка пузырьком (Bubble Sort)

Этот метод постепенно сравнивает соседние элементы массива и меняет их местами, если они находятся в неправильном порядке. Повторяется до тех пор, пока массив не будет полностью отсортирован. Сортировка пузырьком, также известная как Bubble Sort, является одним из наиболее простых алгоритмов сортировки. Она получила свое название благодаря тому, что самый большой “пузырь” всегда всплывает вверх, за ним следует пузырь поменьше и так далее до самого маленького “пузырька”.

Алгоритм сортировки пузырьком основан на принципе "поднимающегося" или "всплывающего" порядка. Он просматривает список несколько раз, сравнивая пары соседних элементов и меняя их местами, если они находятся в неправильном порядке. Этот процесс повторяется до тех пор, пока весь список не станет отсортированным.

Процесс сортировки пузырьком можно описать следующим образом:

* начинаем сравнение первого и второго элементов списка. Если первый элемент больше второго, меняем их местами. Если они уже в правильном порядке, оставляем их как есть;
* Переходим к следующей паре соседних элементов (2-й и 3-й) и сравниваем их. Снова меняем их местами, если они не в правильном порядке.
* Проходим по всему списку, сравнивая и меняя местами соседние элементы, пока не достигнем конца списка. В результате этого первый наибольший элемент "всплывает" на правильное место.
* Повторяем этот процесс для всех элементов списка, кроме последнего, таким образом, что каждый раз на правильное место "всплывает" следующий наибольший элемент.
* После полного прохода по списку и выполнения всех необходимых перестановок, список будет отсортирован в возрастающем порядке. Сортировка пузырьком имеет сложность времени O(n^2), где n - количество элементов в списке.

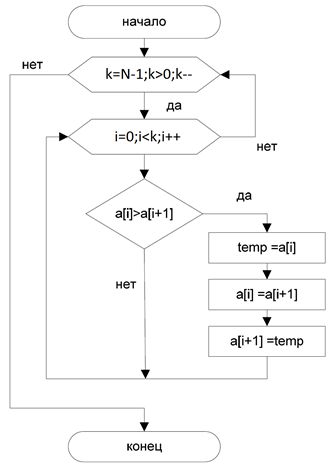


Рисунок 1 – Блок схема алгоритма сортировка пузырьковым методом

* 1. Сортировка вставками (Insertion Sort):

Этот метод является одним из простейших алгоритмов сортировки. Он основан на принципе вставки элемента на правильное место в уже отсортированную часть массива.

Основные шаги алгоритма:

* Начинаем с первого элемента массива и считаем его отсортированным.
* Берем следующий элемент и сравниваем его с предыдущими элементами.
* Если текущий элемент меньше предыдущего, меняем их местами.
* Повторяем шаг 3, пока не найдется правильное место для текущего элемента.
* Переходим к следующему элементу и повторяем шаги 3-5 для всех оставшихся элементов.
* После прохода по всем элементам массив будет отсортирован.

Сортировка вставками имеет время выполнения O(n^2), где n - количество элементов в массиве. В худшем случае алгоритм будет выполнять n(n-1)/2 сравнений и такое же количество перестановок.

Достоинства сортировки вставками включают простую реализацию, работу по месту (не требует дополнительной памяти) и эффективность для небольших списков или уже отсортированных списков. Однако, алгоритм неэффективен для больших списков или списков, в которых элементы находятся в обратном порядке.

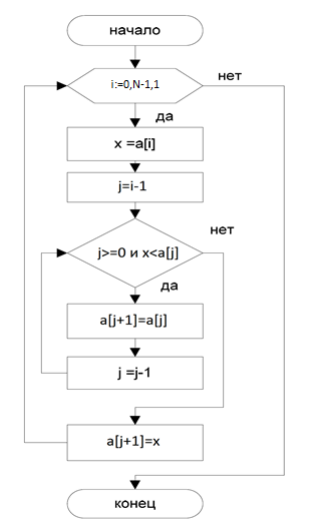


Рис. 2. Блок схема алгоритма сортировки вставками

* 1. Сортировка выбором (Selection Sort):

Этот метод является одним из наиболее простых алгоритмов сортировки и основана на принципе выбора наименьшего (или наибольшего) элемента из оставшейся неотсортированной части массива и его перемещения в начало (или конец) отсортированной части.

Алгоритм сортировки выбором следующий:

* На первом шаге рассматриваем весь массив и находим наименьший элемент.
* Меняем его местами с элементом, стоящим на первой позиции.
* Теперь рассматриваем подмассив, начинающийся с позиции 2 (второй элемент массива) и находим наименьший элемент.
* Меняем его местами с элементом, стоящим на второй позиции.
* Продолжаем эту процедуру для оставшейся неотсортированной части массива, каждый раз выбирая наименьший элемент и перемещая его в начало оставшейся неотсортированной части.
* Повторяем шаги 1-5, пока весь массив не будет отсортирован.

Преимущества сортировки выбором:

* Простота реализации;
* Эффективность на небольших массивах;
* Работает эффективно, когда нужно отсортировать только несколько элементов.

Недостатки сортировки выбором:

* Медленность. Сортировка выбором характеризуется квадратичным временем работы, что делает её неэффективной на больших массивах;
* Неустойчивость. Если в массиве присутствуют одинаковые элементы, после сортировки их относительный порядок может измениться.

Сортировка выбором является наиболее простой сортировкой из всех сортировок и имеет очевидные ограничения по производительности. Тем не менее, она остается полезной при работе с небольшими массивами или в качестве иллюстрации базовых принципов сортировки.

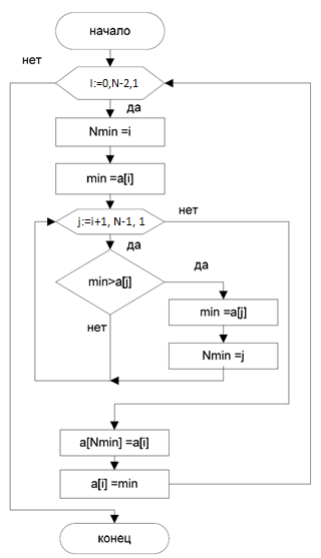


Рис. 3. Блок схема алгоритма сортировки выбором

* 1. **Сортировка Шелла (Shell Sort**):

Этот метод является эффективным алгоритмом сортировки, который был разработан в 1959 году Дональдом Шеллом. Этот алгоритм улучшает производительность других сортировок, таких как сортировка пузырьком и сортировка вставками, путем сокращения размера интервалов сортировки.

Сортировка Шелла использует идею разбиения массива на подмассивы и сортировки каждого из них по отдельности. Для выполнения этого алгоритма мы начинаем с большого интервала и постепенно сокращаем его, пока не достигнем интервала 1. Затем производится окончательная сортировка с шагом 1.

Процесс сортировки Шелла можно разделить на следующие шаги:

* Выбирается интервал, также называемый шагом. Этот интервал может быть задан различными способами, но наиболее распространенным подходом является использование последовательности Хиббарда, которая задается рекуррентным соотношением h = 2h + 1, где h - шаг. Начальное значение h должно быть меньше размера массива.
* Для каждого интервала выполняется сортировка вставками. Это значит, что элементы в каждом интервале сравниваются и переставляются в нужное место, чтобы массив был отсортирован постепенно.
* После сортировки всех интервалов, значит, что массив поделен на несколько частей. Затем выполняется окончательная сортировка с шагом 1, используя обычную сортировку вставками.

Преимущества сортировки Шелла заключаются в следующем:

* Ускорение процесса сортировки по сравнению с другими алгоритмами, такими как сортировка вставками.
* Хорошая производительность для средних и больших массивов данных.
* Возможность сортировки массивов, состоящих из любого типа данных.

Важно понимать, что сортировка Шелла не является стабильным алгоритмом сортировки, что означает, что она не сохраняет порядок элементов с одинаковыми значениями. Также стоит отметить, что выбор интервала является важным фактором, который может влиять на производительность сортировки.

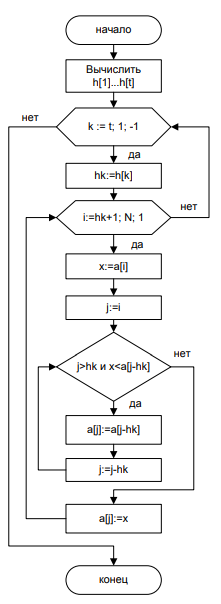


Рис. 4. Блок схема алгоритма сортировка методом Шелла

* 1. Сортировка расческой (Comb Sort):

Этот метод является улучшенной версией сортировки пузырьком. Он использует большие шаги сравнения и обмена элементов, чтобы устранить "черепахи" (маленькие значения, находящиеся в конце массива).

Сортировка расчёской называется так из-за того, что мы как бы расчёсываем массив сначала широким гребнем (большой шаг), потом гребнем поменьше (шаг поменьше). В конце шаг равен единице, как в пузырьковой сортировке.

Эти элементарные методы сортировки просты в реализации, однако их эффективность ниже, чем у более сложных алгоритмов, таких как сортировка слиянием или быстрая сортировка. Сортировка расческой (Comb Sort) является улучшенной версией пузырьковой сортировки.

Принцип работы сортировки расческой заключается в сравнении и обмене элементов на заданной разности, называемой "шагом". При каждой итерации этот шаг уменьшается на фактор, который обычно называется фактором уменьшения. Как только размер шага достигает 1, алгоритм завершается, и это последний проход пузырьковой сортировки.

Алгоритм сортировки расческой выполняется следующим образом:

* Инициализируем шаг равным длине массива.
* Пока шаг больше 1, повторяем следующие шаги: a. Проходим по массиву сравнивая элементы, расположенные на расстоянии шага. Если текущий элемент больше следующего, меняем их местами. b. Уменьшаем шаг на заданное значение фактора уменьшения (обычно 1.3).
* После завершения цикла из пункта 2, выполняем обычную пузырьковую сортировку с шагом 1, чтобы окончательно отсортировать оставшиеся элементы.

Одной из особенностей сортировки расческой является то, что она позволяет более эффективно сортировать массивы, в которых элементы отсортированы в начале или конце. Это связано с тем, что пузырьковая сортировка обычно эффективна, когда элементы перемешаны случайным образом, но не эффективна для уже отсортированных или почти отсортированных массивов.

Плюсы сортировки расческой:

* Быстрее пузырьковой сортировки на больших массивах данных.
* Шаг уменьшения динамически выбирается, что позволяет сортировке быть более эффективной.

Минусы сортировки расческой:

* Худшая производительность для массивов данных, которые уже почти отсортированы.
* Неустойчивость, то есть, при сортировке массивов, содержащих повторяющиеся элементы, относительный порядок повторов может быть изменен.

В целом, сортировка расческой является надежным и эффективным алгоритмом сортировки, способным обрабатывать большие массивы данных в различных сценариях сортировки.

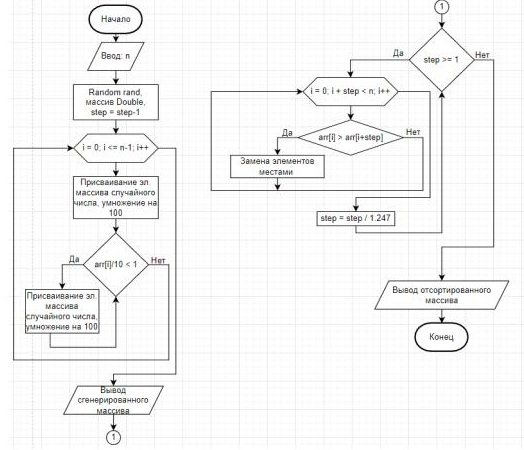


Рис. 5. Блок схема алгоритма сортировка расческой