## **Что такое сортировка**

Сортировка — это процесс **упорядочивания элементов списка** по определённому критерию, например:

* По возрастанию или убыванию чисел (grade, age)
* По алфавиту (name)
* По сложным условиям (например, сначала возраст, потом средний балл)

**Зачем нужна сортировка:**

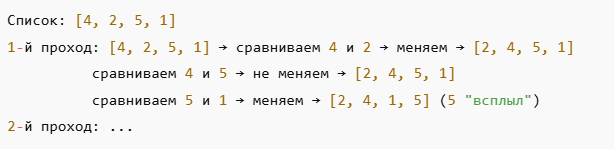
1. Быстрый поиск (например, бинарный поиск работает только на отсортированных списках).
2. Удобный вывод данных для пользователя.
3. Структурирование информации в программе (например, список студентов по успеваемости).
4. Основа для более сложных алгоритмов (слияние, дерево, хеш-таблицы).

## **Типы простых сортировок**

### 1. ****Пузырьковая сортировка (Bubble Sort)****

**Идея:**

* Проходим по списку, сравниваем **соседние элементы**.
* Если они не в правильном порядке, меняем местами.
* После каждого прохода **самый большой элемент "всплывает"** в конец списка.
* Повторяем, пока список полностью не отсортируется.



**Характеристики:**

* Простая реализация.
* Время работы:
  + Лучший случай (уже отсортирован) — O(n)
  + Средний и худший случай — O(n²)
* Пространство: O(1) — сортировка делается **на месте**.

**Плюсы:**

* Очень наглядна для обучения.
* Не требует дополнительных структур данных.

**Минусы:**

* Очень медленная для больших списков.

### 2. ****Сортировка выбором (Selection Sort)****

**Идея:**

* Находим **минимальный (или максимальный) элемент** в списке.
* Ставим его на правильное место в начале (или конце).
* Повторяем для оставшейся части списка.

**Характеристики:**

* Время: O(n²)
* Количество перестановок меньше, чем у пузырьковой.

### 3. ****Сортировка вставками (Insertion Sort)****

**Идея:**

* Разделяем список на **отсортированную и неотсортированную части**.
* Берём элемент из неотсортированной части и вставляем его в нужное место в отсортированной.

**Характеристики:**

* Время: O(n²), но для почти отсортированных списков работает очень быстро.
* Простая реализация для маленьких массивов.

### 4. ****Быстрая сортировка (Quick Sort) и Сортировка слиянием (Merge Sort)****

**Идея:**

* Используют **рекурсию** и деление списка на части.
* Очень эффективны для больших массивов (O(n log n)).

## 🔹 **На что влияет сортировка**

1. **Производительность программы:**
   * Пузырьковая и выбором подходят только для небольших списков (десятки–сотни элементов).
   * Для тысяч элементов лучше использовать Quick Sort или встроенный sorted().
2. **Потребление памяти:**
   * Сортировки на месте (Bubble, Selection, Insertion) не требуют дополнительной памяти.
   * Merge Sort создаёт новые списки → больше памяти.
3. **Сложность кода:**
   * Пузырьковая сортировка — самая простая для понимания и отладки.
   * Quick Sort или Merge Sort сложнее, но быстрее.
4. **Реальные задачи:**
   * Для сортировки студентов, товаров, результатов тестов на небольших списках — пузырьковая или вставками достаточно.
   * Для больших баз данных лучше встроенные функции (sorted) или продвинутые алгоритмы.

**Пузырьковая (Bubble Sort) для нашей задачи**

* + Проходит по списку несколько раз, сравнивает соседние элементы и меняет их местами, если они не в порядке.
  + Простая, но медленная на больших списках (O(n²)).



Код





