



Урок 19. Python: Встроенная сортировка и сравнение сложности

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 19 из 34 · 45 минут · Модуль «Сортировка»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-9-klass-programma>

Записи с одинаковым баллом могут сохранять порядок поступления. При выборе сортировки важны не только числа, но и контракт: изменяется ли исходник и что происходит с равными ключами.

Результат урока: Применить `sorted` и `list.sort`, задать ключ, проверить устойчивость и отличить оценку числа операций от времени запуска.

01. Проблемная ситуация

0–5 минут. Учебные заявки (имя,балл): (B,2),(A,1),(C,2). Нужно по возрастанию балла, при равенстве — в порядке поступления. Запишите ожидаемый порядок имён. Почему обычная сортировка кортежей по первому полю не решит задачу?

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Ключ, новый список и изменение на месте

`sorted(rows, key=...)` возвращает новый список; `rows.sort(key=...)` меняет сам список и возвращает `None`. Ключ `lambda r: r[1]` извлекает балл, поэтому имя не участвует в сравнении. Сортировка Python устойчива: равные ключи сохраняют исходный относительный порядок, в том числе при `reverse=True`.

Сложность описывает рост работы, а не секунды на любом компьютере. Для обычного сравнительного анализа пузырьки в худшем случае делает $n(n-1)/2$ сравнений; сбалансированное слияние при $n=2^k$ имеет k уровней и верхнюю оценку $n \cdot k$ сравнений. Это оценки для названных алгоритмов, не точное число сравнений встроенной функции.

Встроенная сортировка учитывает уже имеющуюся упорядоченность. Для школьного проекта она предпочтительна, когда не требуется реализовать алгоритм вручную. Контракт занятия: до 100 записей (строковое имя, целый балл), без смешивания чисел и строк в одном ключе. Учебные имена условные, личные данные не собираются.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

Из `[("B",2),("A",1),("C",2)]` по баллу получаем A,B,C: B раньше C в исходнике. `reverse=True` даёт B,C,A, а не C,B,A. Последний `print` показывает `None` и уже изменённый `rows`, потому что `sort` не возвращает новый список.

```
rows = [("B", 2), ("A", 1), ("C", 2)]
ordered = sorted(rows, key=lambda r: r[1])
print(ordered)
print(rows)
descending = sorted(rows, key=lambda r: r[1], reverse=True)
print(descending)
result = rows.sort(key=lambda r: r[1])
print(result, rows)
```



Вход	B:2 → A:1 → C:2
По баллу вверх	A:1 → B:2 → C:2
По баллу вниз	B:2 → C:2 → A:1
Равный ключ 2	B остаётся перед C

Устойчивость относится к равным ключам, а не ко всему списку.

03. Практика 1. Проверяем материал

Проверяем контракт библиотеки

14–25 минут. Запускайте код и его варианты для пустого и одноэлементного списков. Числа в последней строке вычислите отдельно, это не измерение времени.

Проверка	Ожидаемое наблюдение
sorted для примера	A,B,C; исходник B,A,C
sort для примера	Возвращает None; исходник A,B,C
[]; [("Q",0)]	[]; одна запись без изменений
n=8; n=16	Пузырёк: 28;120. n log2 n: 24;64.

2 минуты — прогноз четырёх print, 4 — три запуска, 3 — расчёт таблицы сложности, 2 — журнал «операция / возвращаемое значение / исходник». Критерии: верно различены sorted и sort, сохранён порядок B,C, оценки не названы секундами.

04. Практика 2. Перенос способа

Добавляем вторичный ключ

25–33 минуты. Даны [("C",2),("B",2),("A",1),("D",3)]. Теперь нужно сначала больший балл, а при равенстве имя по возрастанию. За 2 минуты составьте составной ключ, за 4 выполните код ниже с заменённым ключом, за 2 объясните отличие от reverse=True по одному баллу. Критерии: порядок D,B,C,A и оба направления.

```
rows = [("C",2), ("B",2), ("A",1), ("D",3)]
# Replace the key with the required two-field key.
print(sorted(rows, key=lambda r: r[1]))
```

05. Обсуждение и границы вывода

33–40 минут. «Встроенная сортировка быстрее на моём коротком списке, значит любой её запуск быстрее любого самописного». За 2 минуты найдите ошибку обобщения, за 3 предложите честное сравнение, за 2 выберите решение для проекта.

06. Индивидуальная проверка

40–45 минут. Дан a=[3,1,3]. Назовите b после b=sorted(a), а после этого, r после r=a.sort() и a после sort. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Первый print: [(A,1),(B,2),(C,2)], второй: исходные B,A,C, третий: B,C,A, четвёртый: None и A,B,C (в реальном выводе строки заключены в кавычки). $8*7/2=28$, $16*15/2=120$; $8*3=24$, $16*4=64$. Рост разных оценок различается, но по двум маленьким тестам нельзя доказать общую сложность библиотеки.

Практика 2: разбор и критерии

Замените ключ на `lambda r: (-r[1], r[0])`. Получится [(D,3),(B,2),(C,2),(A,1)]. Минус меняет направление числового балла, а имя остаётся возрастающим. Один `reverse=True` по баллу дал бы D,C,B,A, сохраняя исходный порядок C,B при равенстве.

Обсуждение: возможный вывод

Время зависит от длины, структуры входа, реализации и условий измерения. Сравнивают один и тот же набор данных и результат, много повторов, без печати внутри измеряемого участка; отдельно анализируют рост операций. Для проекта выбираем встроенную сортировку за ясный контракт и проверенность, а не за обещание победы на каждом микротесте.

Индивидуальная проверка: ответ и критерии

`b=[1,3,3]`; после `sorted a=[3,1,3]`; `r=None`; после `sort a=[1,3,3]`. По одному баллу.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Python 3.11+ и IDLE или редактор; запуск `python lesson.py` (либо `python3 lesson.py`). Внешние библиотеки не нужны. Установку и проверку запуска учитель выполняет заранее. Все программы, данные и ожидаемые ответы даны; регистрация детей и интернет на занятии не нужны.

Уровень — начинающий программист, 9 класс. Предпосылки: списки, кортежи, функции, пузырьёк и слияние. В паре автор прогнозирует и вводит код, проверяющий ведёт журнал, затем роли меняются. Без компьютера выполните трассировку полного кода на бумаге и сопоставьте с готовыми результатами; учитель демонстрирует подготовленный запуск. Бумажный вариант не считается фактическим запуском ученика. Перед практикой учитель выдаёт готовый файл разобранного примера, чтобы дети тратили время на изменение и проверку, а не длинное перепечатывание. Матрица программы предназначена учителю.

Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ



07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Алгоритмы сортировки	Проверяем контракт библиотеки	Применить <code>sorted</code> и <code>list.sort</code> , задать ключ, проверить устойчивость и отличить оценку числа операций от времени запуска.
Структурирование решения и отладка	Добавляем вторичный ключ	Исходник, журнал обычных и граничных тестов, самостоятельное объяснение

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-9-klass-programma>

Python: встроенная сортировка, ключи и устойчивость

<https://docs.python.org/3/howto/sorting.html>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 18.10.2026.