

# Вычислительная сложность алгоритмов

# Зачем это всё?

Недостаточно написать код, который работает, в продакшене нужно писать код который работает эффективно

# Что значит эффективно?

Эффективность мы, да и все в мире меряют по 2 показателям:

- по времени выполнения алгоритма
- по памяти

# Что такое эффективность?

Алгоритм 1:

- взять число
- умножить на 2
- вывести число

Алгоритм 2:

- взять число
- умножить на 2
- прибавить 7
- разделить на 3
- вычесть 4
- умножить на 9
- вывести число

Алгоритм 1 эффективнее алгоритма 2, верно?

# Что такое эффективность?

Алгоритм 1:

- взять число
- умножить на 2
- вывести число

Алгоритм 2:

- взять число
- умножить на 2
- прибавить 7
- разделить на 3
- вычесть 4
- умножить на 9
- вывести число

Алгоритм 1 эффективнее алгоритма 2, верно?

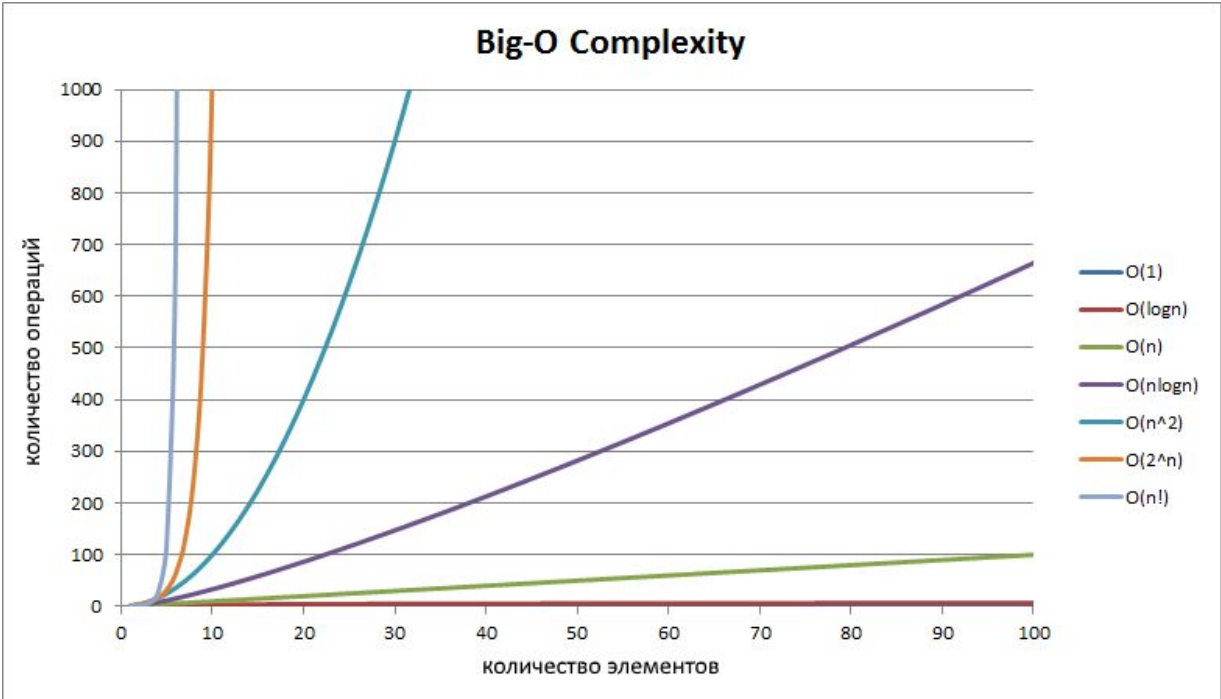
Но, в программировании эти алгоритмы имеют одинаковую сложность

# Сложность предыдущих алгоритмов

Она равна постоянному числу

# Как же так?

# Всё дело в том, как в программировании измеряют сложность



# Примеры алгоритмов

Быстро

$O(1)$  – сложение чисел

$O(\log(n))$  – бинарный поиск

$O(n)$  – поиск числа в списке

$O(n^2)$  – пузырьковая сортировка

$O(2^n)$  – перебор всех подмножеств множества

$O(n!)$  – задача коммивояжера с использованием полного перебора всех возможных путей



Медленно