

Алгоритмы и структуры данных на Python

Урок 4

## Эмпирическая оценка алгоритмов на Python

Измерения времени работы с использованием timeit. Профайлер.

#### Вопросы

- 1. По каким критериям можно оценить алгоритм?
- 2. Способы оценки алгоритма



#### Цели урока

- Изучить методы оценки алгоритма
- Изучить средства Python для оценки алгоритмов



#### План урока

- Эмпирическая оценка алгоритмов
- Оценка сложности алгоритма
- Примеры для практического закрепления



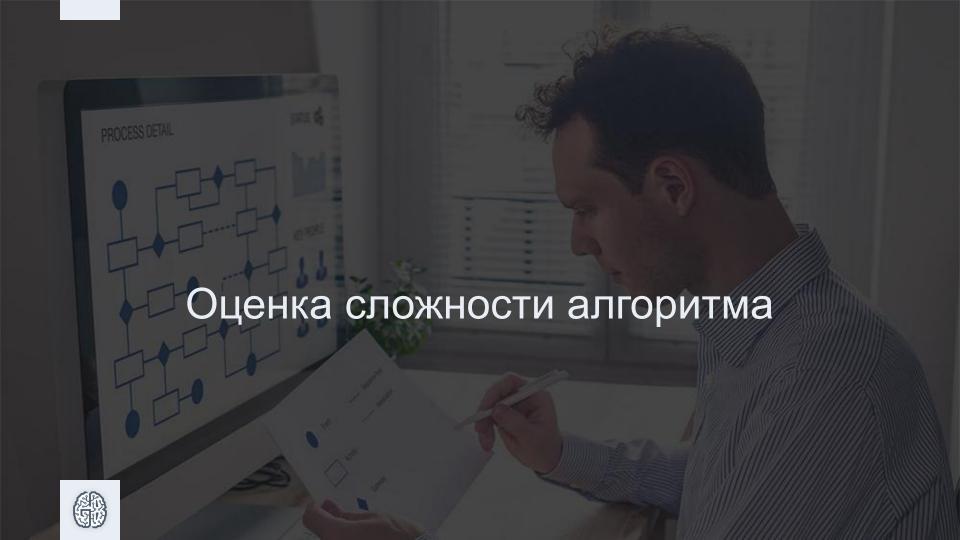
### Эмпирическая оценка алгоритмов



#### Эмпирическая оценка алгоритмов

- Измерения времени работы с использованием **timeit**
- Профайлер





#### Оценка сложности алгоритма

O(n) — линейная сложность O(n2) — квадратичная сложность

O(log n) логарифмическая сложность



#### Асимптотический анализ

#### Константный О(1)

```
def GetCount(items):
return items.Length;
```



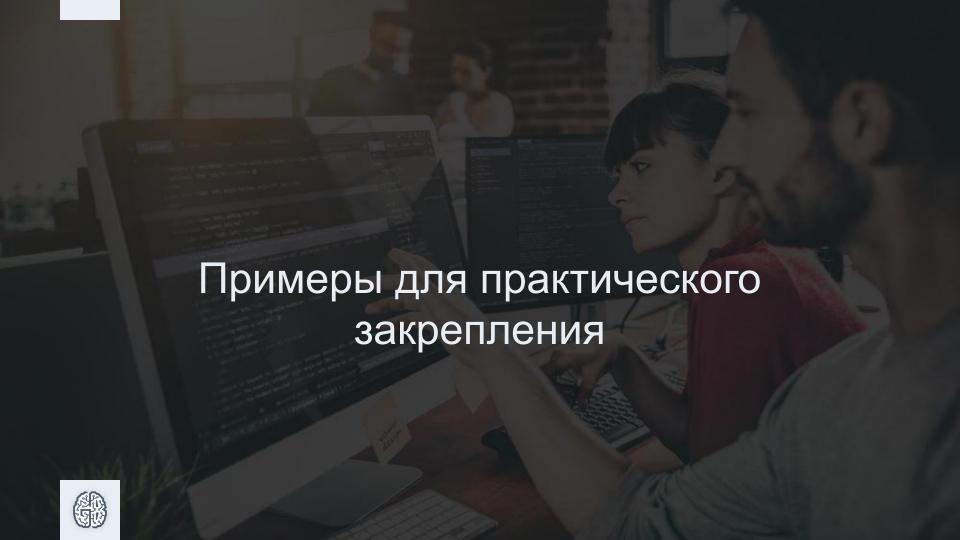
#### Асимптотический анализ

Логарифмический – O(log n)

Квадратичный – O(n 2)

Линеарифметический – O(n·log n)





# Задача 1. Оптимизация алгоритма на примере вычисления чисел Фибоначчи



#### Практическое задание

1. Проанализировать скорость и сложность одного любого алгоритма, разработанных в рамках практического задания первых трех уроков.

Примечание: попробуйте написать несколько реализаций алгоритма и сравнить их.



#### Практическое задание

- 2. Написать два алгоритма нахождения i-го по счёту простого числа.
  - Без использования решета Эратосфена;
  - С использованием решета Эратосфена.



