

# Алгоритмы

June 11, 2021

## 1 Сложность, тестирования и особые случаи

### 1.1 Сложность алгоритма

**Сложность алгоритма** - порядок количества действий, которые выполняет алгоритм

1. Константы не включаются в  $O(N)$
2. Асимптотически константы не влияют на скорость работы алгоритма при больших параметрах
3. *Пространственная сложность* - количество использованной памяти
4. Требуется решение с наименьшей асимптотической сложностью

#### 1.1.1 Задача UTF-8

Задача:

Дана строка в кодировке UTF-8, найти самый частый символ.

Решение #1:

1. Внешний цикл перебирает все позиции
2. Внутренний цикл для каждой позиции перебирает все другие символы и считает совпадения
3.  $O(N^2)$ , память:  $O(N)$

Решение #2

1. Перебираем все символы в строке и находим уникальные буквы
2. Потом перебираем уже уникальные буквы а не все
3.  $O(kN) \sim O(N)$ , память:  $O(N + K)$

Решение #3:

1. Создаем словарь, ключ - символ, а значение - сколько раз он встречался
2. Если символ встречается впервые, то инициализируем нулем
3. Прибавляем к элементу словаря с ключом, совпадающим с этим символом, единицы

4.  $O(N + k)$ , память:  $O(K)$

## 1.2 Тестирование

Что нужно тестировать:

1. Тесты из условия (если есть)
2. Общие случаи
3. Особые случаи

Советы по составлению тестов:

1. Если есть примеры - решить руками и сверить ответ
2. Проверить последовательность из одного элемента и пустую последовательность
3. "Краевые эффекты" - проверить, что программа работает корректно в начале и в конце последовательности