

# Поиск Абелевых строк наибольшей длины

И. Збань

Научный руководитель: В. Аксёнов



УНИВЕРСИТЕТ ИТМО

6 июня 2017 г.

# Постановка задачи

Задача: Нахождение наибольшей общей Абелевой подстроки (НОАП) и поиск Абелевых подквадратов.

# Постановка задачи

Задача: Нахождение наибольшей общей Абелевой подстроки (НОАП) и поиск Абелевых подквадратов.

Две строки Абелево эквивалентны, если одна строка получается из другой перестановкой символов.

# Постановка задачи

Задача: Нахождение наибольшей общей Абелевой подстроки (НОАП) и поиск Абелевых подквадратов.

Две строки Абелево эквивалентны, если одна строка получается из другой перестановкой символов.

Абелев подквадрат — подстрока, представимая как конкатенация двух Абелево эквивалентных строк.

- ▶ Быстроразвивающаяся область, много публикаций за последнее время

- ▶ Быстроразвивающаяся область, много публикаций за последнее время
- ▶ Актуальность: подзадачи в бионформатике (gene clusters), фильтры в задаче поиска образца с ошибками

- ▶ Быстроразвивающаяся область, много публикаций за последнее время
- ▶ Актуальность: подзадачи в бионформатике (gene clusters), фильтры в задаче поиска образца с ошибками
- ▶ Близость с известной задачей 3SUM

# Содержание работы

В работе выполнены следующие части:

- ▶ Реализация и оценка эффективности теоретического алгоритма для решения 3SUM+ для монотонных множеств на примере задачи о числе абелевых подквадратов



# Содержание работы

В работе выполнены следующие части:

- ▶ Реализация и оценка эффективности теоретического алгоритма для решения 3SUM+ для монотонных множеств на примере задачи о числе абелевых подквадратов
- ▶ Анализ задачи НОАП для бинарного алфавита

# Содержание работы

В работе выполнены следующие части:

- ▶ Реализация и оценка эффективности теоретического алгоритма для решения  $3SUM+$  для монотонных множеств на примере задачи о числе абелевых подквадратов
- ▶ Анализ задачи НОАП для бинарного алфавита
- ▶ Решение задачи НОАП для произвольного алфавита

# Подсчёт числа Абелевых подквадратов

Задача о числе Абелевых подквадратов сводится к  $3SUM^+$   
( $A + B = C$ )

# Подсчёт числа Абелевых подквадратов

Задача о числе Абелевых подквадратов сводится к  $3SUM^+$   
( $A + B = C$ )

$$A = B = \{(c_a(i), c_b(i))\}, C = \{2 \cdot c_a(i), 2 \cdot c_b(i)\}$$

где  $c_a(i), c_b(i)$  — число букв  $a$  и  $b$  на префиксе длины  $i$

# Подсчёт числа Абелевых подквадратов

Задача о числе Абелевых подквадратов сводится к  $3SUM^+$   
( $A + B = C$ )

$$A = B = \{(c_a(i), c_b(i))\}, C = \{2 \cdot c_a(i), 2 \cdot c_b(i)\}$$

где  $c_a(i), c_b(i)$  — число букв  $a$  и  $b$  на префиксе длины  $i$

и число подстрок —  $(\#3SUM^+(A, B, C) - (n + 1))/2$

# Сравнение алгоритмов на простой строке

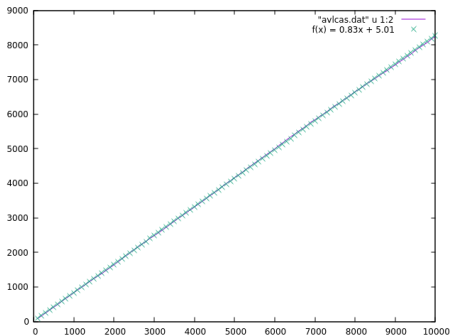
Картиночка на которой видно, что квадрат работает быстро, а 1.86 — медленно

# Сравнение алгоритмов на случайном тесте

Картиночка, на которой видно, что квадрат работает быстро, а 1.86 — оооочень медленно

# НОАП на бинарном алфавите

График зависимости НОАП двух случайных бинарных строк от  $n$ , длины строк. Сгенерирован путем усреднения результата за  $10^4$  испытаний.





# НОАП на бинарном алфавите

В работе доказана оценка сверху, что матожидание длины НОАП двух случайных бинарных строк ограничена сверху линейной функцией с коэффициентом меньше единицы. Тем самым опровергнуто гипотеза из первоисточника

В работе предлагается сведение задачи к 3SUM, лучшее решение которой на данный момент имеет асимптотику  $\mathcal{O}(n^{1.86})$ .

# НОАП на произвольном алфавите

Был разработан алгоритм решения НОАП на произвольном алфавите за  $\mathcal{O}(n^2 \log \sigma)$  времени и  $\mathcal{O}(n)$  памяти.

Сравнение алгоритмов решения НОАП

Год	Авторы	Время	Память
2015	Кто-то	$\mathcal{O}(n^2 \sigma)$	$\mathcal{O}(n \sigma)$
2016	Кто-то	$\mathcal{O}(n^2 \sigma)$	$\mathcal{O}(n)$
2016	SPIRE	$\mathcal{O}(n^2 \log^2 n \log^* n)$	$\mathcal{O}(n \log^2 n)$
2017	Я	$\mathcal{O}(n^2 \log \sigma)$	$\mathcal{O}(n)$

# Краткое описание алгоритма

Идея — построить персистентный массив вектора Парей для строки-конкатенации обеих данных строк, посчитать некий хеш от каждого корня, и проверить, были ли одинаковые версии, соответствующие обеим строкам

# Схема вычисления хеша

Тут какая-то непонятная картинка

# Вопросы?

Спасибо за внимание.