

# Поиск Абелевых строк наибольшей длины

И. Збань

Научный руководитель: В. Аксёнов



УНИВЕРСИТЕТ ИТМО

19 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.

## Определение 1

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.

## Определение 1

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

## Определение 2

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

- ▶ Быстроразвивающаяся область, много публикаций за последнее время

- ▶ Быстроразвивающаяся область, много публикаций за последнее время
- ▶ Встречается как: подзадача в бионформатике (gene clusters, mass indexing), фильтр в задаче поиска образца с ошибками

- ▶ Быстроразвивающаяся область, много публикаций за последнее время
- ▶ Встречается как: подзадача в бионформатике (gene clusters, mass indexing), фильтр в задаче поиска образца с ошибками
- ▶ Связь 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

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  $i$ .

Смысл сведения можно понять, исходя из того, что мы ищем такую подстроку  $[i; j]$  что верно  $c_a(k) - c_a(i) = c_a(j) - c_a(k)$ , и так же по второму символу, где  $k$  — центр подстроки,  $(i + j)/2$ .

# Подсчёт числа Абелевых подквадратов

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

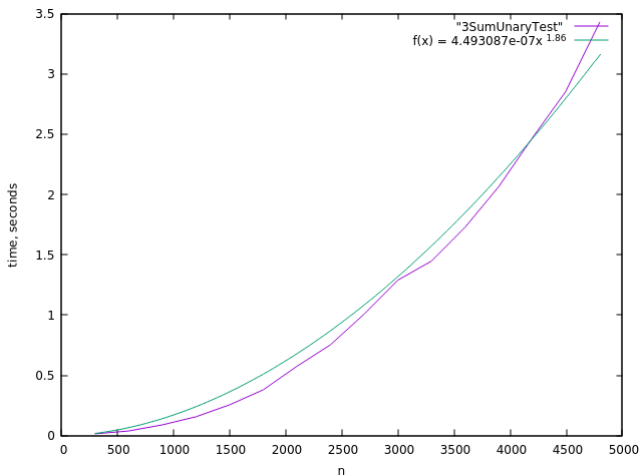
$$A = B = \{(c_a(i), c_b(i))\}, C = \{2 \cdot c_a(i), 2 \cdot c_b(i)\}$$

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

Смысл сведения можно понять, исходя из того, что мы ищем такую подстроку  $[i; j)$  что верно  $c_a(k) - c_a(i) = c_a(j) - c_a(k)$ , и так же по второму символу, где  $k$  — центр подстроки,  $(i + j)/2$ .

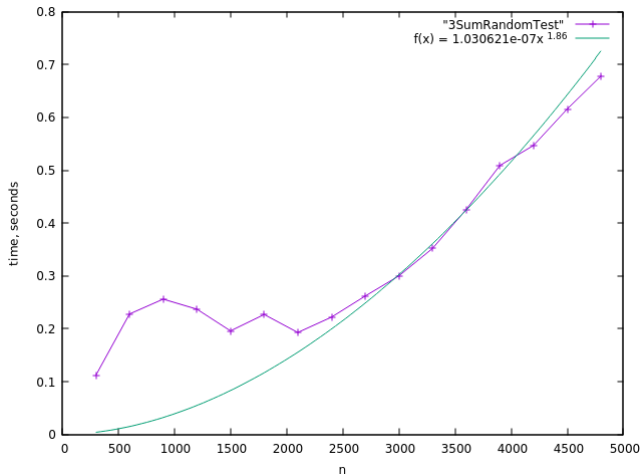
Искомое число подстрок —  $(\#3SUM^+(A, B, C) - (n + 1))/2$

# Анализ алгоритма на константной строке



Алгоритм работает заметно медленнее, чем решение за  $n^2$ , которое за секунду успевает посчитать около 10,000.

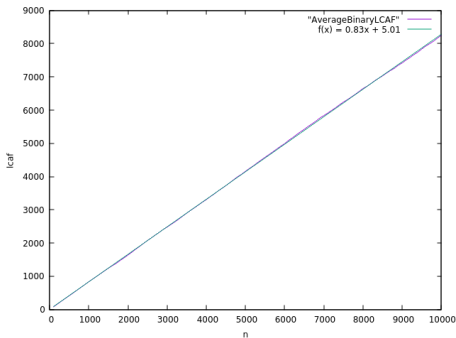
# Сравнение алгоритмов на случайном тесте



Алгоритм работает снова заметно медленнее, чем решение за  $n^2$ .

# 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 опровергнута гипотеза из первоисточника.

Так же в работе доказана оценка снизу линейной функцией  $0.05n$ .

В работе предлагается сведение задачи к  $\text{monotonic 3SUM+}$ , 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	A. Alattabi	$\mathcal{O}(n^2 \sigma)$	$\mathcal{O}(n\sigma)$
2016	S. Grabowski	$\mathcal{O}(n^2 \sigma)$	$\mathcal{O}(n)$
2016	S. Grabowski	$\mathcal{O}(n^2 \log^2 n \log^* n)$	$\mathcal{O}(n \log^2 n)$
2017	Данная работа	$\mathcal{O}(n^2 \log \sigma)$	$\mathcal{O}(n)$

# Краткое описание алгоритма

- ▶ Для фиксированной длины надо проверить, есть ли пара подстрока с одинаковым вектором Парейя. При сдвиге вправо в нем меняются 2 элемента, будем хранить его в персистентном дереве отрезков.

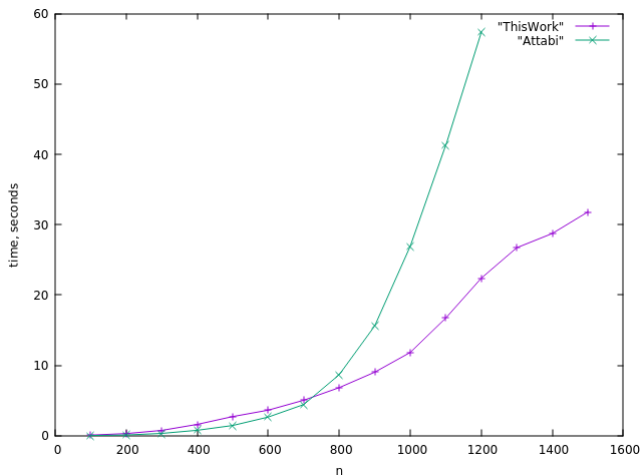
# Краткое описание алгоритма

- ▶ Для фиксированной длины надо проверить, есть ли пара подстрок с одинаковым вектором Парейя. При сдвиге вправо в нем меняются 2 элемента, будем хранить его в персистентном дереве отрезков.
- ▶ Для линейной памяти можно использовать limited node copying.

# Краткое описание алгоритма

- ▶ Для фиксированной длины надо проверить, есть ли пара подстрок с одинаковым вектором Пареля. При сдвиге вправо в нем меняются 2 элемента, будем хранить его в персистентном дереве отрезков.
- ▶ Для линейной памяти можно использовать limited node copying.
- ▶ Можно пересчитывать уникальный хеш-индекс каждой вершины (в том числе не созданных явно), все еще используя линию памяти.

# Экспериментальные результаты алгоритма



Предложенный алгоритм работает заметно быстрее предложенных ранее.

# Вопросы?

Спасибо за внимание.