Реализация алгебраических алгоритмов приближенного поиска по образцу в сжатых данных

Федоркина М. О.

Научный руководитель: д.ф-м.н. А. В. Тискин

Санкт-Петербургская школа физико-математических и компьютерных наук

НИУ ВШЭ – Санкт-Петербург

Задача приближенного поиска по образцу

Приближенный поиск по образцу (Approximate Pattern Matching)

- Строка-образец p длины m
- Строка-текст t длины $n \geq m$
- Цель: найти все подстроки текста, похожие на образец
- Метрика похожести: наибольшая общая подпоследовательность

Применения:

- Проверка правописания
- Определение спама
- Сравнение последовательностей нуклеотидов

Контекстно-свободные грамматики

Контекстно-свободная грамматика

- Правила вида:
  - $t_k = \alpha$, где α — алфавитный символ
  - $t_k = t_i t_j$, для некоторых i, j, $1 \leq i,j < k$
  - $t_k = \epsilon$, где ϵ — пустая строка

Текст, сжатый контекстно-свободной грамматикой

- Несжатая строка s имеет длину n
- Соответствующая ей КСГ состоит из $\bar{n}$ правил
- Правило $t_{\bar{n}}$ разжимается в s
- Примеры: LZ78, LZW, UNIX-утилита ‘compress’

Мотивация

- Пока не найдено решения задачи приближенного поиска по образцу за время, содержательнее меньше $O(mn)$
- $O(mn)$ это медленно на практике, поэтому в реальности используются различные эвристики и решения для частных случаев
  - алгоритмы, находящие ответ приближенно
  - алгоритмы, работающие для специальных видов задачи, например маленькое m
- Предлагается ускорить приближенный поиск с помощью обработки больших сжатых файлов без распаковки

Обзор работ: несжатые данные

Приближенный поиск на несжатых данных

- UNIX-утилита agrep[1], библиотека TRE[2]
  - различные метрики “похожести” на образец
  - длина образца < 32
- SSEARCH, часть пакета ПО FASTA[3]
  - Для нуклеотидных последовательностей
- Эвристические решения:
  - FASTA, BLAST[4]
  - Python difflib[5]

---

[1] https://github.com/Wikinaut/agrep
[2] https://github.com/laurikari/tre
[3] https://github.com/wrpearson/fasta36
[4] https://blast.ncbi.nlm.nih.gov/
[4] https://docs.python.org/3/library/difflib.html

Обзор работ: сжатые данные

Приближенный поиск на сжатых данных

- Теоретическое решение за $O(\bar{n}m + \bar{m}n)$ для двух LZW-сжатых текстов[1]
- Теоретическое решение за $O(\bar{n}m^2 \log m)$[2]
- Теоретическое решение за $O(\bar{n}m \log m)$[3]

---

[1] Maxime Crochemore, Gad M. Landau, and Michal Ziv-Ukelson. “A Subquadratic Sequence Alignment Algorithm for Unrestricted Scoring Matrices.” In: (2003).

[2] Patrick Cégielski et al. “Window Subsequence Problems for Compressed Texts.” In: (2006).

[3] Alexander Tiskin. “Fast Distance Multiplication of Unit-Monge Matrices.” In: (2013).

Цель и задачи

Цель: Реализовать алгебраический алгоритм приближенного поиска по образцу в сжатых данных и проверить его эффективность.

Задачи:

- Реализовать теоретически описанный алгоритм для эффективного приближенного поиска по образцу в сжатых данных за время $O(\bar{n}m \log m)$
- Адаптировать алгоритм для работы с использующимися на практике форматами, например LZ78, LZW и UNIX Z-compress
- Сравнить время работы алгоритма с имеющимися алгоритмами приближенного поиска

Детали реализации: общая идея

Структура	Пара строк	Унимонжева матрица	Перестановка
Операция	Конкатенация (над НОП)	Тропическое умножение матриц	Умножение в 0-моноиде Гекке симметрической группы

- В основе алгоритма лежит изоморфизм между некоторыми алгебраическими структурами
- Для быстрого подсчета НОП реализовано умножение за $O(n \log n)$ над перестановками
- Со строками и унимонжевыми матрицами не работаем явно, реализованы отображения между структурами и все операции в коде проводятся с перестановками

Детали реализации: поиск НОП

- “Разделяй-и-властвуй”, внутри конкатенация с помощью умножения за $O(n \log n)$ в 0-моноиде Гекке симметрической группы
- Алгоритм “разделяй-и-властвуй”,
  - Работаем с *подперестановками* — подмножеством элементов перестановки
  - Конкатенация: проход двумя парами указателей по подперестановкам
  - Для прохода: следующий по порядку строк/столбцов ненулевой элемент за $O(1)$
  - Храним значения: количество ненулевых элементов левее и выше, правее и ниже указателя
- Время работы $O(mn)$, рекурсивная структура полезна для обработки сжатых данных

Детали реализации: сжатые данные

- Сравнение несжатой строки-образца p и строки-текста t, сжатой контекстно-свободной грамматикой за $O(m\bar{n}\log m)$
- Рекурсия по КС-грамматике, использующая модификацию рекурсивного алгоритма подсчета НОП, реализация никак не зависит от формата сжатия
- Для обработки алгоритма сжатия необходим декомпрессор, за линейное от длины файла время восстанавливающий структуру грамматики по сжатому файлу
- Сейчас поддерживаются:
  - LZ78
  - LZW
  - UNIX-compress

Сравнение

При сравнении:

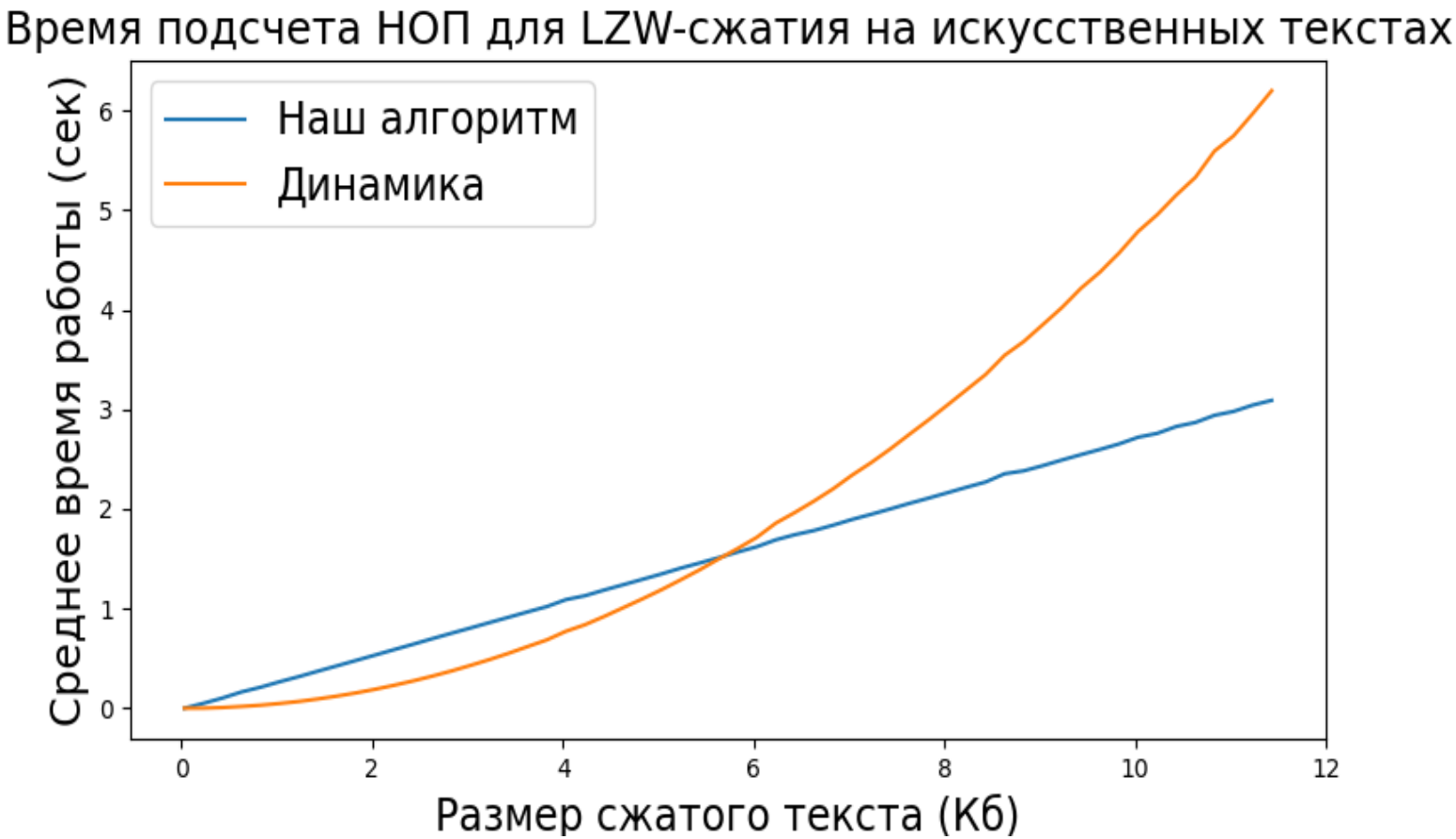
- Смотрим на среднее время работы по 20 запускам на образцах и текстах одинаковой длины
- Образцы длины 16, случайно сгенерированы

Сравниваем время работы алгоритма с:

- стандартным ДП для приближенного поиска на строках
- для compress можно удобно сравниваться с существующими утилитами для приближенного поиска на файлах, распаковывая формат .Z утилитой decompress
  - agrep — наиболее используемое решение не для специфического формата данных

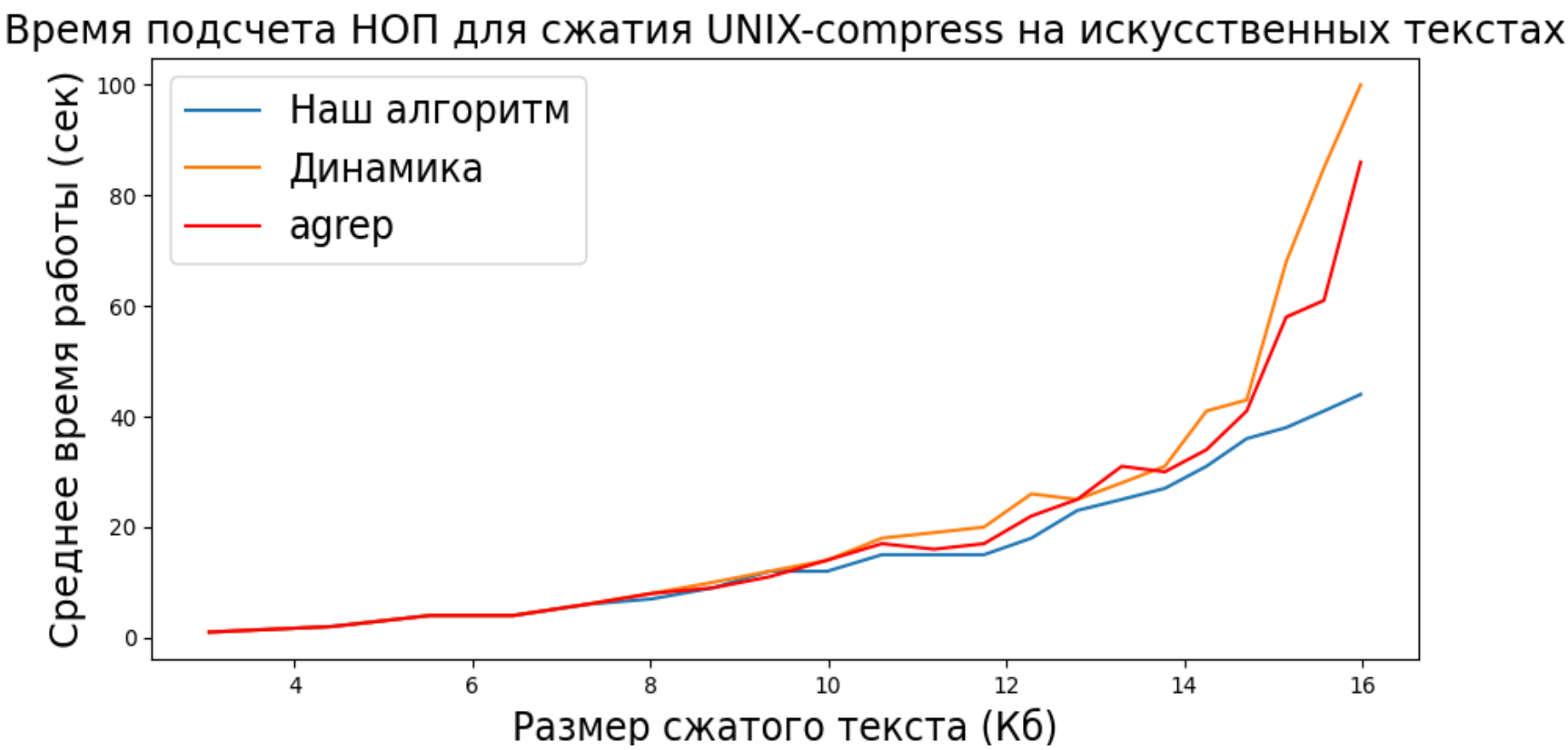
Сравнение: LZW, фиктивные данные

- Искусственные тексты фиксированной длины, сгенерированные так, чтобы достигалось квадратичное сжатие



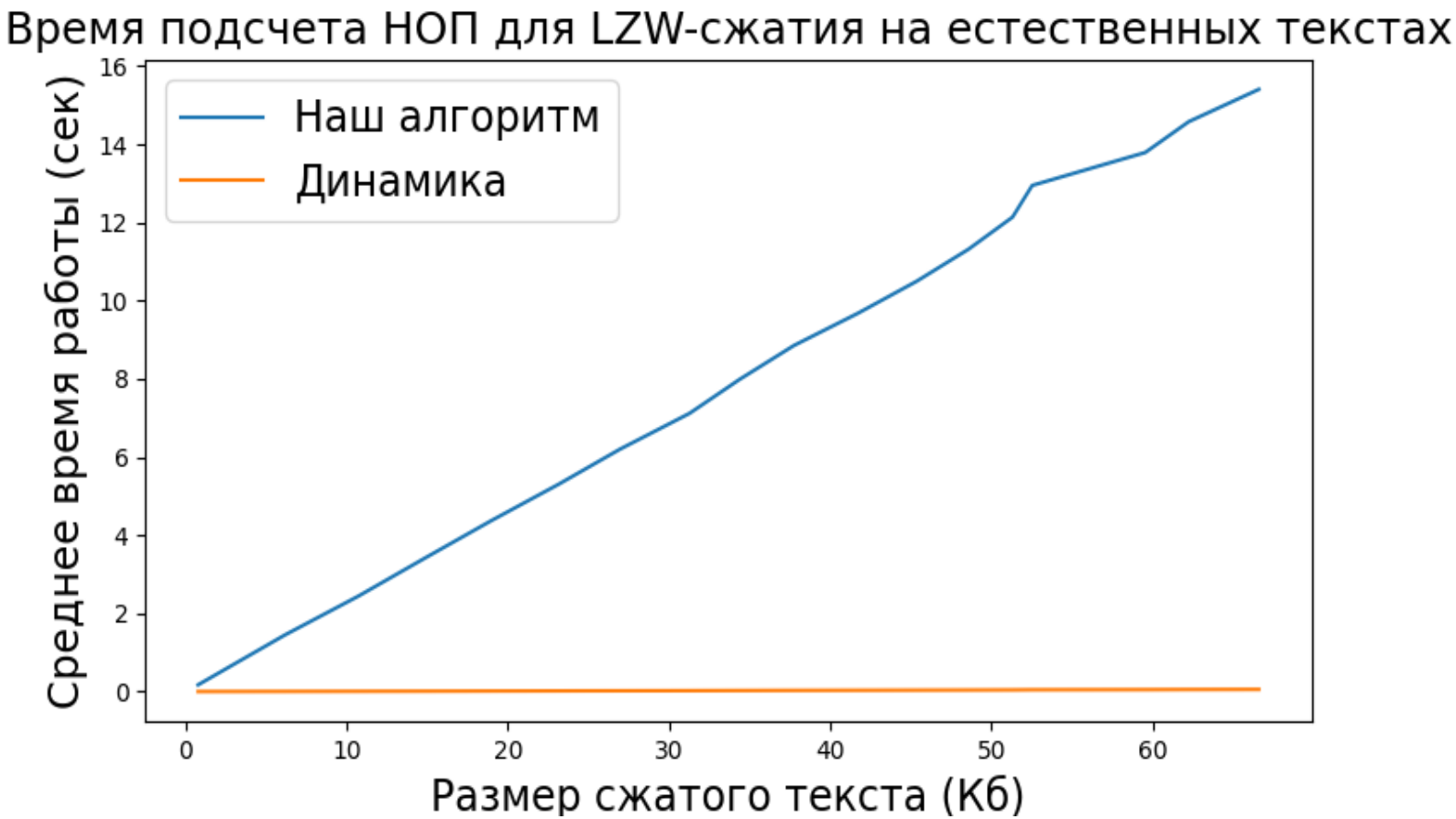
Сравнение: compress, фиктивные данные

- Утилитам compress, agrep необходим запуск в Unix-подобной ОС
- Для измерения времени работы экспериментов и их запуска был написан bash-скрипт



Сравнение: LZW, реальные данные

- Тексты с Project Gutenberg
- Предобработка: токенизация, лемматизация, удаление стоп-слов



Результаты

- Реализован алгоритм для приближенного поиска по образцу в данных, сжатых контекстно-свободной грамматикой
- Улучшение времени работы приближенного поиска по образцу в специально сгенерированных текстовых строках, сжатых LZW и LZ78, по сравнению с ДП
- Улучшение времени работы приближенного поиска по образцу в специально сгенерированных текстовых файлах, сжатых UNIX-compress, по сравнению с ДП и agrep
- Не показано улучшение времени работы приближенного поиска по образцу в данных, используемых на практике
- Часть работы представлена на конференции Polynomial Computer Algebra’2020