На правах рукописи

Рубинчик Михаил Валентинович

Вычислительная сложность некоторых задач обработки строк

01.01.09 — дискретная математика и математическая кибернетика

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата физико-математических наук

Екатеринбург 2016 г.

Работа выполнена на кафедре алгебры и дискретной математики Института математики и компьютерных наук Уральского федерального университета имени первого Президента России Б. Н. Ельцина, г. Екатеринбург.

Научный руководитель:	доктор физико-математических наук, профессор Шур Арсений Михайлович
Официальные оппоненты:	Фрид Анна Эдуардовна, доктор физико-математических наук, преподаватель-исследователь университета Экса-Марселя (Aix Marseille University) Куликов Александр Сергеевич, кандидат физико-математических наук, научный сотрудник ФГБУН «Санкт-Петербургское отделение Математического института им. В. А. Стеклова РАН»
Ведущая организация:	ФГАОУ ВО «Национальный Исследовательский Университет «Высшая Школа Экономики»

Защита диссертации состоится «25» мая 2016 г. в 17:00 на заседании диссертационного совета Д 002.202.02 на базе ФГБУН «Санкт-Петербургское отделение Математического института им. В. А. Стеклова РАН» (191023, г. Санкт-Петербург, ул. наб. р. Фонтанки, 27).

С диссертацией можно ознакомиться в библиотеке и на сайте ФГБУН «Санкт-Петербургское отделение Математического института им. В. А. Стеклова РАН» http://www.pdmi.ras.ru/

Автореферат разослан «   » ________ 2016 г.

Ученый секретарь
диссертационного совета
доктор физ.-мат. наук Малютин Андрей Валерьевич

Общая характеристика работы

Актуальность темы исследования. Строка (последовательность символов) — один из центральных объектов компьютерных наук. Любые данные, имеющие линейную структуру (например, текст на естественном или искусственном языке, лог сервера или колебания курса валюты) можно представить в виде строки и изучать особенности в этой строке. Начало активного изучения алгоритмов, связанных со строками, приходится на 1970е годы. Раздел компьютерных наук, посвящённый изучению алгоритмов обработки строк, называется *стрингологией* (stringology). Впервые это название было предложено в 1985 в работе [11], когда область уже имела какой-то набор результатов, но сложно было назвать её отдельной наукой. Однако уже в 90х годах наблюдается стремительный рост числа публикаций по данной тематике. Одной из самых сильных предпосылок развития является появление биоинформатики, науки, изучающей «биологические строки» (ДНК, РНК, белки и др.). Именно биоинформатика породила множество задач, изучаемых стрингологами. Не менее важным источником задач являются современные информационные технологии, требующие эффективных алгоритмов анализа и обработки данных. Среди задач этой группы выделяются задача поиска в массиве данных и задача сжатия данных. Кроме того, стрингология, относимая к фундаментальной информатике, решает большое число чисто математических задач, не нашедших еще практического применения. Сюда относится, в частности, классификация задач о строках по их вычислительной сложности. Многие теоретические задачи приходят в стрингологию из комбинаторики слов.

Развитость стрингологии как самостоятельной науки показывает, в частности, наличие ряда монографий [1, 6, 7, 24] и специализированных международных конференций и семинаров: String Processing and Information Retrieval (SPIRE), Combinatorial Pattern Matching (CPM), Prague Stringology Conference (PSC), StringMasters.

Данная работа сконцентрирована на двух группах задач из стрингологии. Первая — алгоритмы, связанные с поиском и анализом палиндромов* в строках. Вторая — алгоритмы, связанные с повреждёнными строками.

Методы исследования. В данной работе большинство результатов — это алгоритм, либо структура данных. Для каждого алгоритма

*Палиндром — строка, равная своему развороту.

или структуры приведено описание, скорость работы и используемая память, а также доказательство корректности и оценок скорости работы и используемой памяти. Остальные результаты — доказательство принадлежности задачи к тому или иному классу сложности. В диссертации использованы различные методы теории алгоритмов, теории вычислительной сложности, стрингологии и комбинаторики слов. Разработан новый метод решения задач, связанных с палиндромами, основанный на предложенной автором эффективной структуре данных для хранения информации о палиндромах.

Чтобы строго определить скорость алгоритма и используемую им память, следует обратиться к понятию модели вычислений. В данной работе используется модель Word-RAM, являющаяся одной из самых популярных и хорошо приближающая реальные компьютерные вычисления. В этой модели используемая память — это массив машинных слов (фактически, целых чисел) с прямым доступом, а основные операции над ячейками, включающие арифметические, логические и операции чтения/записи, производятся за константное время. Мы будем полагать, что символы алфавита являются целыми числами от 0 до M, где M не более чем в константное число раз превосходит размер входа. Входные данные для любой задачи представляют собой строку с последовательным доступом для посимвольного чтения в память.

Если задача решается *оффлайн*, решающему алгоритму разрешено вначале выполнить чтение всей входной строки, а затем приступить к обработке и выдаче результатов. В случае решения *онлайн*, условия значительно более жесткие: после прочтения очередного символа алгоритм должен выдать ответ для прочитанной к настоящему моменту строки до того, как читать следующий символ. Таким образом, задача решается для всех префиксов входной строки. Такая модель особенно важна в алгоритмах, использующихся на практике, когда, например, к серверу поступает множество запросов и данные нужно обновлять «на лету», а не выполнять сразу много запросов «оптом».

Далее мы говорим об оффлайновых и онлайновых алгоритмах в зависимости от вида решения, которое они дают.

Цели и задачи. Целями диссертации являются классификация ряда комбинаторных задач о строках по их вычислительной сложности, а также построение теоретически и практически эффективных алгоритмов решения этих задач.

Задачами диссертации являются:
— построение алгоритмов низкой вычислительной сложности для ряда задач о палиндромах в строках
— классификация задач об оптимальном восстановлении поврежденных строк по их вычислительной сложности и построение эффективных алгоритмов для полиномиально разрешимых вариантов этих задач.

Рассматриваются следующие основные задачи о палиндромах.
— подсчет различных подпалиндромов: дана строка S, найти число различных палиндромов, являющихся подстроками в S;
— подсчет палиндромно-насыщенных строк: даны числа n и k, найти для всех $i \leq n$ число k-ичных строк длины i, содержащих ровно i (максимальное количество) различных непустых палиндромов;
— разбиение на палиндромы: даны строка S и число k, ответить, представима ли S в виде конкатенации ровно k палиндромов;
— палиндромная длина: дана строка S, найти минимальное k, для которого S является конкатенацией k палиндромов.

Поврежденная строка над алфавитом Σ — это строка над расширенным алфавитом $\Sigma \cup \Gamma$, где каждый «поврежденный» символ из Γ «совместим» с некоторыми символами из Σ. Восстановление поврежденной строки состоит в замене всех поврежденных символов в строке на совместимые символы из Σ. Рассматриваются две задачи об оптимальном восстановлении.
— Даны две поврежденные строки, более длинный «текст» T и более короткий «шаблон» P, требуется восстановить T и P так, чтобы максимизировать число вхождений P в T.
— Даны две поврежденные строки, более длинный «текст» T и более короткий «шаблон» P, требуется восстановить T и P так, чтобы минимизировать сумму расстояний Хэмминга[†] между P и всеми подстроками той же длины в T.

Положения, выносимые на защиту. В рамках решения поставленных задач получены

1. новая структура данных «овердрево» («eertree») для быстрого решения различных задач о палиндромах в строках, алгоритмы по-

[†] *Расстоянием Хэмминга* между строками S и T одинаковой длины называется число $d(S,T) = |\{i \mid S[i] \neq T[i]\}|$.

строения этой структуры и ее специализированных версий [30]

2. два алгоритма решения задачи о числе различных палиндромов в строке [28, 30]

3. два алгоритма решения задачи о разбиении строки на заданное число палиндромов [29, 30]

4. алгоритм решения задачи о разбиении строки на минимально возможное число палиндромов [30]

5. алгоритм перечисления палиндромно-насыщенных строк [30]

6. алгоритмы решения частных случаев и доказательство NP-трудности общего случая задачи о восстановлении повреждённых строки и шаблона с максимизацией числа вхождений [27]

7. алгоритмы решения частных случаев и доказательство NP-трудности общего случая задачи о восстановлении повреждённых строки и шаблона с минимизацией суммарного расстояния Хэмминга [27]

Научная новизна. Все результаты диссертации являются новыми.

Теоретическая и практическая ценность. Диссертация носит теоретический характер. Результаты могут быть использованы для дальнейших научных исследований в области алгоритмов на строках, в частности, по сформулированным в диссертации открытым вопросам, связанным с палиндромами и повреждёнными строками. Также можно применять полученные результаты при обучении алгоритмам. Например, структура данных «овердрево» может предварять тему «суффиксное дерево». Овердрево уже сейчас входит в программы курсов и спецкурсов по алгоритмам ряда российских вузов, в том числе СПб АУ, МФТИ, ВШЭ. Кроме того, все алгоритмы и структуры данных, предложенные в диссертации, просто и эффективно реализуются в виде программного кода, что делает их практически применимыми, например, для постановки вычислительных экспериментов.

Апробация результатов работы. Все представленные результаты опубликованы в [27–30]. Вклад соавторов в совместные работы таков: Шуру А.М. принадлежит постановка задачи и оптимизация некоторых доказательств в работах [28–30], Гамзовой Ю.В. принадлежит постановка задачи и общая методика исследования в работе [27], Косолобову Д.А.

принадлежит нижняя оценка в работе [28], а также основной алгоритм и доказательство его корректности и эффективности в работе [29]. Автору принадлежат алгоритмы и доказательства основных результатов в работах [27,30], основной алгоритм в [28] и первоначальная конструкция алгоритма из [29].

Также по теме диссертации опубликовано трое тезисов [31–33].

Результаты, приведенные в диссертации, докладывались на международной конференции по комбинаторным алгоритмам (IWOCA 2015), на российско-финском симпозиуме по дискретной математике (RuFiDim 2012), на днях стрингологии в Лондоне/Лондонском алгоритмическом семинаре (LSD & LAW 2016), на днях компьютерных наук в Екатеринбурге (CSEdays2013), а также на семинарах «Алгебраические системы» (УрФУ, рук. Шеврин Л. Н., 2015–2016) и «Дискретная математика» (УрФУ, рук. Шур А.М., 2011–2016).

Структура диссертации. Диссертация состоит из трёх глав, заключения и списка литературы. Объём диссертации составляет 83 страницы, библиография включает 55 наименований.

Краткое содержание работы

В **главе 1** диссертации приводятся определения и некоторые дополнительные сведения, необходимые для понимания всего дальнейшего текста, и в концентрированном виде излагаются сами обсуждаемые результаты. Основные результаты распределены по главам 2 и 3.

Глава 2 посвящена изучению палиндромов в строках. Большая её часть содержит описание новой структуры данных под названием «овердрево»[‡] и её применения к решению ряда задач о палиндромах.

Известно, что число различных непустых подпалиндромов в строке длины n не превосходит n (см. [8]) и известен линейный от n оффлайновый алгоритм нахождения их всех (см. [14]). Нахождение оптимального онлайнового алгоритма решения той же задачи — открытая проблема, поставленная в [14]. В разделе 2.2 описан онлайновый алгоритм, решающий эту проблему за время $O(n \log |\Sigma|)$, основанный на классических алгоритмах Укконена [25] и Манакера [20]. После чего приведём существенно более простой и быстрый алгоритм с такой же асимптотикой, обобщением которого получена новая структура данных «овердрево».

[‡]Название отражает то, что это древовидная структура для хранения и обработки палиндромов.

Овердрево строки S (см. рис. 1) представляет собой орграф, вершины которого — все палиндромы, встречающиеся в S. Ребра, помеченные алфавитными символами, соединяют пары палиндромов вида (Q, aQa). Кроме того, каждый палиндром соединен непомеченным ребром — «суффиксной ссылкой» — со своим максимальным суффикс-палиндромом. Построение овердрева, которое может быть осуществлено за время $O(n \log |\Sigma|)$ онлайн и $O(n)$ оффлайн, немедленно отвечает на вопрос о числе различных подпалиндромов в S. В оставшейся части раздела доказываются свойства овердрева и демонстрируются возможности его применения на примерах сложных задач студенческих олимпиад по программированию.

В разделе 2.3 приведён ряд естественных модификаций овердрева.

Первая из них — совместное овердрево нескольких строк, позволяющее эффективно решать задачи о сравнении палиндромной структуры строк. Далее приводятся модификации, использующие «дополнительные» суффиксные ссылки, за счет которых удалось значительно уменьшить время, требуемое в худшем случае для одной итерации построения овердрева. Таким образом удалось эффективно построить овердрево с «откатами», поддерживающее, наряду с добавлением символа, операцию удаления последнего символа за константное время. Далее овердрево с откатами используется для перечисления палиндромно-насыщенных строк.

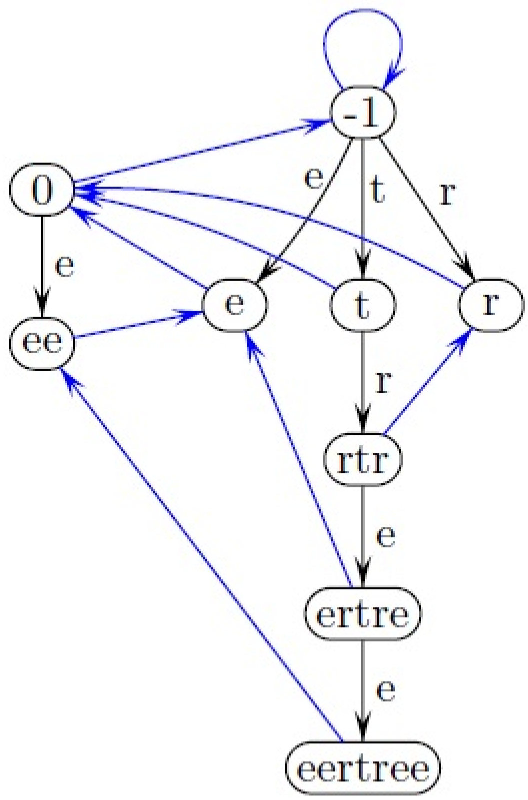

Рис. 1: Овердрево для «eertree». Стрелки, помеченные символом — рёбра, остальные — суффиксные ссылки.

Структура палиндромно-насыщенных строк изучается в комбинаторных работах начиная с [8]. В [13] впервые был поставлен вопрос об оценке числа таких строк. Изложенный далее в разделе 2.3 алгоритм, основанный на овердреве с откатами, перечисляет все такие строки до заданной длины над заданным алфавитом константного размера за время $O(1)$ на одну строку. Полученные результаты вошли в онлайн-энциклопедию целочисленных последовательностей, см. [23, A216264] и были использованы в работе [15] для оценки функции роста множества бинарных палиндромно-насыщенных строк.

Заключительная часть раздела 2.3 посвящена построению на основе овердрева персистентной структуры данных. Персистентное овердрево поддерживает множество версий входной строки и позволяет отвечать на запросы (в том числе о модификации) к любой из версий. На добавление символа к существующей версии тратится логарифмический объем времени и памяти, характерный для эффективных персистентных структур.

Раздел 2.4 посвящён изучению разбиений строки на палиндромы. Задачи о разбиении на палиндромы представляют давний интерес в теории формальных языков. Известно, что асимптотически наилучший универсальный алгоритм распознавания контекстно-свободных языков — алгоритм Валианта [26] — работает за более чем квадратичное время. При этом ни для одного контекстно-свободного языка до сих пор не удавалось доказать несуществование алгоритма, распознающего этот язык за линейное время. Одно время предполагалось, что языки конкатенации чётных палиндромов и язык конкатенации палиндромов с длиной больше единицы невозможно распознать за $O(n)$ (см. [19, раздел 6]). Оказалось, что это не так: в [19] описан линейный алгоритм для распознавания первого языка, а в [12] — для второго. Распознавание языка конкатенации k палиндромов для произвольного фиксированного k оказалось гораздо более сложной задачей. В [12] приведён линейный алгоритм, распознающий язык для $k = 1, 2, 3, 4$. Вариант такого алгоритма можно найти в [7, раздел 8]. В [12] и [7] был поставлен вопрос о существовании линейного алгоритма для $k > 4$. В статье [29] приведён алгоритм Косолобова, работающий за $O(kn)$, который, к сожалению, важен только для теории ввиду своей реализационной сложности и большой константы. В разделе 2.4 диссертации приведён алгоритм, основанный на «овердреве», работающий за $O(n \log n)$, но имеющий маленькую константу, не зависящую от k, и решающий одновременно задачу нахождения палиндромной длины.

Алгоритм опирается на ряд комбинаторных лемм о «сериях» палиндромов. Заметим, что немного раньше был опубликован другой алгоритм поиска палиндромной длины за $O(n \log n)$ [9], но наш алгоритм имеет значительно меньшие константы по времени и памяти, а также более простой код. Раздел 2.4 заканчивается гипотезой о возможности линейного решения задач о разбиении на палиндромы и результатами, поддерживающими эту гипотезу, в частности, оффлайновым алгоритмом, строящим овердрево за линейное время.

В **главе 3** рассматривается задача об оптимальном восстановлении повреждённых строк. Решаются два варианта этой задачи, различающиеся критерием оптимальности, который в обоих случаях связан с поиском подстроки в строке. Для обоих вариантов приведены полиномиальные решения в частных случаях и доказательства NP-трудности для общего случая.

Задача поиска подстроки в строке — это одна из самых известных и хорошо изученных задач стрингологии. Ее стандартная постановка состоит в следующем. Дано две строки — более длинный «текст» S и более короткий «шаблон» P, требуется найти все вхождения шаблона в текст, т. е. все пары (i, j) такие, что подстрока в S, начинающаяся в i-й и заканчивающаяся в j-й позиции, равна P. Эта задача имеет большое прикладное значение для информационного поиска и биоинформатики. Существует множество вариаций данной задачи (например, с ограничениями на используемую память или на доступ к строкам, с приблизительным поиском, с «неточным» шаблоном, и т.п.).

Одно из наиболее интересных с практической точки зрения обобщений задачи поиска подстроки в строке состоит в предположении, что текст и/или шаблон могут быть повреждены при передаче данных (например, из-за шума в канале, или при распознавании печатного текста, или при точечных мутациях, если речь идет о биологических «строках», таких как цепочки ДНК). повреждённые символы нельзя идентифицировать однозначно, но для каждого из них можно указать множество символов исходного алфавита, из которых в результате повреждения мог получиться данный символ. Таким образом, каждому символу алфавита повреждённых строк поставлено в соответствие некоторое множество символов исходного алфавита, т.е. между этими алфавитами определено бинарное отношение — *отношение совместимости.* Две строки совместимы, если их длины равны и буквы, стоящие в одинаковых позициях, совместимы.

Важным частным случаем повреждённых строк являются *частичные* строки, содержащие повреждённые символы только одного вида — «джокеры», совместимые со всеми символами алфавита. Задачи поиска для повреждённых строк хорошо известны и более сложны, чем для обычных строк; см., например, [5,10,22]. Кроме того, модель повреждённых символьных последовательностей изучается в комбинаторике слов, см., например, [18], а работы по частичным словам составляют достаточно большой массив, см., например, [2–4, 16, 17, 21].

В главе 3 решаются две задачи, в которых поиск играет вспомогательную роль, а цель — восстановить повреждённый текст и повреждённый шаблон оптимальным образом.

В разделе 3.3 рассмотрена задача восстановления, в которой требуется максимизировать число вхождений шаблона в текст. В нём рассмотрены полиномиальные решения задачи для случая неповреждённого текста и неповреждённого шаблона, а также приведено доказательство, что задача NP-трудна в общем случае даже для бинарного алфавита.

Раздел 3.4 описывает результаты по задаче восстановления с минимизацией суммарного расстояния Хэмминга. В нём приведены полиномиальные решения для случая неповреждённого текста либо шаблона, для случая частичных строк и циклического текста, а также для бинарного алфавита. Также в нём приведено доказательство NP-трудности задачи в общем случае и сформулирована гипотеза о том, что для частичных строк задача остается NP-трудной. Для удобства, результаты главы 3 сведены в таблицу:

Вариант	Задача 1	Задача 2
Неповреждённый текст	Поиск повреждённого шаблона + суффиксный массив, $O(n\|\Sigma\|\log n)$ для повреждённых и $O(n \log n)$ для частичных	Жадный алгоритм, $O(n+m\|\Sigma\|)$ для повреждённых $O(n+m\log\|\Sigma\|)$ для частичных
Неповреждённый шаблон	Динамическое программирование + префикс-функция, $O(n\log n+mt)$ для частичных, $O(n\|\Sigma\|\log n+mt)$ для повреждённых	Жадный алгоритм, $O(n\|\Sigma\|)$ для повреждённых $O(n+m\log\|\Sigma\|)$ для частичных
Бинарный алфавит	NP-трудна	Минимальный разрез
Частичные строки	NP-трудна	NP-трудна (гипотеза)
Частичные строки с циклическим текстом	NP-трудна	Жадный алгоритм, $O(n)$
Циклический текст	NP-трудна	NP-трудна
Общий случай	NP-трудна	NP-трудна

В **заключении** кратко даётся резюме по проделанной работе, указываются направления дальнейших исследований и формулируется ряд оставшихся открытыми вопросов.

Благодарности. Автор благодарит Юлию Васильевну Гамзову, своего первого научного руководителя, привившего вкус к научным исследованиям, Арсения Михайловича Шура, руководителя данной работы, внёсшего большой вклад в постановку многих задач, получение новых результатов и в написание статей. А также Дмитрия Косолобова за совместную работу над статьями, подтолкнувшую к получению других результатов, Григория Назарова, Олега Меркурьева и Александра Кулькова за ценные обсуждения.

Список литературы

[1] Гасфилд Д. Строки, деревья и последовательности в алгоритмах. — Невский диалект СПб., 2003.

[2] Шур А. М., Гамзова Ю. В. Частичные слова и свойство взаимодействия периодов // Известия Российской академии наук. Серия математическая. — 2004. — Т. 68, № 2. — Р. 191–214.

[3] Berstel J., Boasson L. Partial words and a theorem of Fine and Wilf // Theoret. Comput. Sci. — 1999. — Vol. 218. — P. 135–141.

[4] B. Blakeley, F. Blanchet-Sadri, J. Gunter, N. Rampersad. On the complexity of deciding avoidability of sets of partial words // Theor. Comput. Sci. — 2010. — Vol. 411, no. 49. — P. 4263–4271.

[5] Clifford P., Clifford R. Simple deterministic wildcard matching // Information Processing Letters. — 2007. — Vol. 101, no. 2. — P. 53–54.

[6] Crochemore M., Hancart C., Lecroq T. Algorithms on strings. — Cambridge University Press, 2007.

[7] Crochemore M., Rytter W. Jewels of stringology. — World Scientific Publishing Co. Pte. Ltd., 2002.

[8] Droubay X., Justin J., Pirillo G. Episturmian words and some constructions of de Luca and Rauzy // Theoretical Computer Science. — 2001. — Vol. 255. — P. 539–553.

[9] G. Fici, T. Gagie, J. Kärkkäinen, D. Kempa. A subquadratic algorithm for minimum palindromic factorization // J. of Discrete Algorithms. — 2014. — Vol. 28. — P. 41–48.

[10] Fischer M., Paterson M. String matching and other products // SIAM-AMS Proceedings. — 1974. — Vol. 7. — P. 113–125.

[11] Galil Z. Open problems in stringology // Combinatorial Algorithms on Words. — Springer, 1985. — P. 1–8.

[12] Galil Z., Seiferas J. A Linear-Time On-Line Recognition Algorithm for "Palstar" // J. ACM. — 1978. — Vol. 25, no. 1. — P. 102–111.

[13] A. Glen, J. Justin, S. Widmer, L. Zamboni. Palindromic richness// European J. of Combinatorics. — 2009. — Vol. 30. — P. 510–531.

[14] Groult R., Prieur E., Richomme G. Counting distinct palindromes in a word in linear time // Information Processing Letters. — 2010. — Vol. 110. — P. 908–912.

[15] Guo C., Shallit J., Shur A. M. On the combinatorics of palindromes and antipalindromes. — arXiv:1503.09112 [cs.FL]. — 2015.

[16] V. Halava, T. Harju, T. Kärki, P. Séébold. Overlap-freeness in infinite partial words// Theoret. Comput. Sci. — 2009. — Vol. 410, no. 8-10. — P. 943–948.

[17] Idiatulina L. A., Shur A. M. Periodic partial words and random bipartite graphs // Fundam. Inform. — 2014. — Vol. 132, no. 1. — P. 15–31.

[18] Kärki T., Harju T., Halava V. Interaction properties of relational periods // Discrete Mathematics & Theoretical Computer Science. — 2008. — Vol. 10, no. 1.

[19] Knuth D. E., Morris J. H., Pratt V. R. Fast pattern matching in strings // SIAM J. on Computing. — 1977. — Vol. 6. — P. 323–350.

[20] Manacher G. A new linear-time on-line algorithm finding the smallest initial palindrome of a string // J. ACM. — 1975. — Vol. 22, no. 3. — P. 346–351.

[21] Manea F., Mercaş R. Freeness of partial words // Theoret. Comput. Sci. — 2007. — Vol. 389, no. 1-2. — P. 265–277.

[22] Muthukrishnan S., Ramesh H. String Matching Under a General Matching Relation // Proc. 12th Conference on Foundations of Software Technology and Theoretical Computer Science. — Vol. 652 of LNCS. — Berlin : Springer-Verlag, 1992. — P. 356–367.

[23] Sloane, N.J.A.: The on-line encyclopedia of integer sequences. — 1964–2016. — Available at `http://oeis.org`.

[24] Smyth W. Computing patterns in strings. — Pearson Education, 2003.

[25] Ukkonen E. On-line construction of suffix trees // Algorithmica. — 1995. — Vol. 14, no. 3. — P. 249–260.

[26] Valiant L. G. General context-free recognition in less than cubic time // J. of Computer and System Sciences. — 1975. — Vol. 10, no. 2. — P. 308–314.

Публикации автора в изданиях, рекомендованных ВАК

[27] Рубинчик М. В., Гамзова Ю. В. Две задачи о восстановлении поврежденных строк // Сибирские электронные математические известия. — 2013. — Т 10. — С. 538–550.

[28] Kosolobov D., Rubinchik M., Shur A. M. Finding distinct subpalindromes online // Proc. Prague stringology conference 2013 / Ed. by Jan Holub, Jan Žďárek. — CTU, 2013. — P. 63–69.

[29] Kosolobov D., Rubinchik M., Shur A. M. Pal^k is linear recognizable online // SOFSEM 2015: Theory and Practice of Computer Science. — Springer-Verlag Berlin Heidelberg, 2015. — Vol. 8939 of LNCS. — P. 289–301.

[30] Rubinchik Mikhail, Shur Arseny M. EERTREE: An Efficient data structure for processing palindromes in strings // Combinatorial algorithms: Proc. IWOCA 2015. — Vol. 9538 of LNCS. — Springer International Publishing, 2016. — P. 321–333.

Другие публикации автора

[31] Kosolobov D., Rubinchik M. An optimal online algorithm for finding all distinct subpalindromes of a string // Algorithms & Complexity. Abstracts of Reports and Other Materials of the 6th School "Computer Science Days in Ekaterinburg" A. S. Kulikov, A. M. Shur (eds.). — Ekaterinburg: Ural University Press, 2013. — P. 44–46.

[32] Rubinchik M., Gamzova Y.V. Two pattern matching problems for strings with compatibility relations // V. Halava, J. Karhumaki, Y. Matiyasevich (Eds.), Proc. 2nd Russian Finnish Symposium on Discrete Mathematics (RuFiDiM II). — Vol. 17 of TUCS Lecture Notes. — Turku Centre for Computer Science, 2012. — P. 151–152.

[33] Rubinchik Mikhail, Shur Arseny M. On the number of distinct subpalindromes in words // Proc. 3rd Russian Finnish Symp. on Discrete Mathematics. Inst. Appl. Math. Research, Petrozavodsk. — 2014. — P. 96–98.