РАЗВИТИЕ ИНТЕЛЛЕКТА ШКОЛЬНИКОВ

С. М. Окулов

АЛГОРИТМЫ ОБРАБОТКИ СТРОК

2-е издание

Москва
БИНОМ. Лаборатория знаний

УДК 519.85(023)
ББК 22.18
О-52

Серия основана в 2008 г.

Окулов С. М.

О-52 Алгоритмы обработки строк / С. М. Окулов. — 2-е изд. — М. : БИНОМ. Лаборатория знаний, 2015. — 256 с. : ил. — (Развитие интеллекта школьников).

ISBN 978-5-9963-1931-2

На материале задачи поиска подстроки в строке, решению которой посвящены работы многих профессионалов за последние 20–30 лет, показано, как построить занятия по информатике, чтобы побудить школьника к творчеству, развить у него вкус к решению исследовательских проблем.

Для школьников, преподавателей информатики, а также для студентов, выбравших информатику в качестве основной специальности. Книга может быть использована как в обычных школах при проведении факультативных занятий, так и в образовательных учреждениях с углубленным изучением информатики и математики.

УДК 519.85(023)
ББК 22.18

Учебное издание

Серия: «Развитие интеллекта школьников»

Окулов Станислав Михайлович

АЛГОРИТМЫ ОБРАБОТКИ СТРОК

Ведущий редактор *Д. Ю. Усенков*
Художник *Н. А. Новак*
Технический редактор *Е. В. Денюкова*
Корректор *Л. Н. Макарова*
Компьютерная верстка: *В. А. Носенко*

Подписано в печать 20.11.14. Формат 60×90/16.
Усл. печ. л. 16,00.

Издательство «БИНОМ. Лаборатория знаний»
125167, Москва, проезд Аэропорта, д. 3
Телефон: (499) 157-5272
e-mail: binom@Lbz.ru
http://www.Lbz.ru, http://e-umk.Lbz.ru, http://metodist.Lbz.ru

---

ISBN 978-5-9963-1931-2

Оглавление

Михаилу, сыну моему, посвящаю.

Предисловие

Все должно быть изложено так
просто, как только возможно, —
но не проще.

Альберт Эйнштейн

Что может быть эффективнее для развития творческих возможностей школьника и его интеллекта, чем решение задач, казалось бы, очень простых, но «тянущих» за собой проблемы, исследованием которых занимались ведущие специалисты по информатике в последние 20–30 лет? Одной из таких задач является *задача поиска подстроки в строке*, которая так или иначе затрагивается в любом учебнике по информатике. Длительность ее решения с помощью самого простого алгоритма пропорциональна произведению длин строки и подстроки, и, несмотря на возросшую производительность компьютера, она оказывается слишком большой для многих приложений. Можно ли найти такие алгоритмы решения этой задачи, чтобы произведение заменялось хотя бы суммой? Оказывается, да, и эта замена является сутью работ лучших умов в информатике, многие из которых продолжают свою деятельность и в настоящее время.

Данная книга конструктивно построена в виде занятий, ее материал апробирован в ходе проведения реальных уроков[1)]. Особенность изложения этого материала заключается в том, что он не приводится в виде конечных результатов (лемм, теорем, фактов и т. д.). Напротив, автором сделана попытка показать сам процесс получения нового результата, а он не появляется сразу доказательно оформленным. Конеч-

[1)] Первая попытка изложения части предлагаемого здесь материала была сделана в 1997 г. (*Бабушкина И. А., Бушмелева Н. А., Окулов С. М., Черных С. Ю.* Конспекты занятий по информатике (практикум по Турбо Паскалю). — Киров: Изд-во ВГПУ, 1997).

ное оформление по принятым «правилам игры» при написании книг такого типа обычно скрывает способ его «рождения» — в каких «муках» и как он получен. «Ни один ученый не мыслит формулами», — любил говорить Альберт Эйнштейн. В основе любого алгоритма лежит какая-то образная картинка или ясная идея. Воссоздать эту картинку, взять на себя смелость утверждения о «рождении» результата именно так, как описывается в книге (через примеры, через эксперименты — а они обязательны — и ошибки с «набросками» кода) — страшно. Но если читатель подвергнет все сомнению, то автор будет считать, что он уже достиг цели, ибо специалист по информатике обязан все подвергать сомнению и ничего не принимать на веру! Такое сомнение в правильности результатов и самостоятельная работа приведут вас к моментам «Эврика!», к рождению нового, и, может быть, ваша фамилия так же войдет в историю информатики, как и фамилии авторов рассматриваемых здесь алгоритмов...

В предмете «информатика», а именно в олимпиадных соревнованиях по информатике, сложилась уникальная ситуация, которой нет ни в одном школьном предмете. Например, тематика заданий олимпиад по математике опирается на школьный курс, что вполне естественно. В олимпиадной же информатике (назовем ее так) существует как минимум три направления: первое, поддерживаемое и развиваемое международным сообществом, связано с алгоритмами и программированием; второе — это олимпиады по базовому курсу информатики; третье — различного рода соревнования по использованию информационных технологий. Связь первого направления со школьным курсом информатики ограничена несколькими разделами, второе направление полностью адекватно курсу, а третье — лишь частично. Для достижения результатов в рамках первого направления знать и уметь требуется много, очень много, и любой школьный учебник не входит в этот необходимый минимум. Если выразиться точнее, то победитель соревнований первого типа не всегда сможет выразить свои мысли тем языком, который задается учебниками, хотя, конечно, он знает их материал, но только на совершенно другом уровне понимания и осознания.

Рассматриваемые в книге алгоритмы (основные) входят в примерную программу по олимпиадной информатике всего одной строкой — «*алгоритмы поиска подстроки в строке*

за O(n+m)»[1]. Эти алгоритмы относятся к дидактическим единицам, «изучение которых формирует у школьников ключевые умения в области олимпиадной подготовки, открывает перед участником олимпиадного состязания возможность проявить свой творческий потенциал на достойном уровне ... — победителей и призеров заключительных этапов Всероссийской олимпиады школьников»[2].

Структура книги

Первая глава, с одной стороны, является вводной, а с другой — дает основы предварительного анализа строк, без которых понимание многих изложенных далее алгоритмов невозможно. Но, вероятно, главным в ней следует считать «очерчивание» одного из основных способов построения эффективных алгоритмов, который заключается в тщательном анализе исходных данных с целью выявления в них закономерностей, а затем — в использовании этих закономерностей при решении основной задачи.

Во второй главе рассматриваются ставшие уже классикой алгоритмы Д. Кнута – Дж. Морриса – В. Пратта; Р. Бойера – Дж. Мура; Р. Карпа – М. Рабина; *Shift–And* (Б. Дёмёлки – Р. Беза-Йетс – Г. Гоннет); М. Крочемора и М. Мейна – Р. Лоренца. Если первые четыре алгоритма посвящены проблеме поиска подстроки в строке, то последние два являются основополагающими в задаче анализа свойств строки (текста). Во второй главе кратко показано, как использовать аппарат теории автоматов при описании алгоритмов на строках.

Третья глава целиком посвящена *деревьям суффиксов* — структуре данных, в которой фиксируются особенности строки (текста), позволяющие эффективно решать многочисленные задачи обработки строк. Рассмотрены два алгоритма — Э. Укконена и Е. Мак-Крейга, однако их рассмотрению предшествует анализ простых методов построения дерева суффиксов, что позволяет сделать изложение доступным и ясным (с точки зрения автора). На остальные известные алгоритмы решения этой задачи в данной книге приводятся только ссылки.

В четвертой главе рассматриваются задачи вычисления расстояния между строками и нахождения наибольшей общей подпоследовательности двух строк. Анализ первой за-

[1] *Кирюхин В. М.* Информатика: всероссийские олимпиады. — М.: Просвещение, 2008. С. 71.

[2] Там же. С. 67.

дачи ограничивается основным алгоритмом и результатом Э. Укконена – Ю. Майерса. Вторая проблема анализируется через достижения С. Нудельмана – К. Вунша, Д. Ханта – Т. Зиманского и Л. Эллисона – Т. Дикса.

Пятая глава посвящена достаточно значимой задаче данной проблематики — приближенному поиску подстрок в тексте. Даны простые алгоритмы, а также алгоритм С. Ву – Ю. Менбера, алгоритм решения задачи о k-несовпадениях и идейные основы алгоритма Ю. Майерса.

Вместо заключения читателям предлагается небольшое эссе о предмете «информатика», из которого становятся яснее роль и место как материала данной книги, так и результата, получаемого в итоге деятельности по освоению рассматриваемых алгоритмов.

В приложениях раскрываются некоторые моменты организации углубленного экспериментального исследования алгоритмов и приводится ряд проблем (задач) для индивидуальной творческой работы[1)]. Рассмотренные в данной книге алгоритмы — это, если можно так выразиться, только первый «пласт» указанного раздела информатики. Проблемы, приведенные в приложении, лишь частично восполняют этот пробел, полностью устраняемый, вероятно, лишь последующими работами.

Благодарности

Я благодарю коллег: Евгения Вячеславовича Котельникова, Андрея Васильевича Лялина и многих других за интеллектуальную помощь, без которой вряд ли состоялся бы этот труд. Особая признательность — Дмитрию Юрьевичу Усенкову, сотруднику издательства «БИНОМ. Лаборатория знаний», оперативность и доброжелательность работы которого вызывают восхищение. Глубочайшая благодарность — директору издательства Михаилу Николаевичу Бородину, который не только поверил в «пришедшего с улицы» автора (в 2001 г.), но и все эти годы профессионально поддерживает его деятельность.

1) На русском языке имеются только две фундаментальные переводные книги по данной проблематике: *Смит Б.* Методы и алгоритмы вычисления на строках: пер. с англ. — М.: ООО «И. Д. Вильямс», 2006 и *Гасфилд Д.* Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология: пер. с англ. И. В. Романовского. — СПб.: Невский Диалект; БХВ-Петербург, 2003. Автор, конечно же, использовал их (как и англоязычные статьи) при написании данной книги, которая, выражаясь педагогическим языком, является пропедевтикой к изучению названных книг, предоставляющих обширный материал по проблемам для индивидуальной творческой работы как школьника, так и студента.

Глава 1

Строки

И приходят мне в голову сказки
Мудрецами отмеченных дней,
И блуждаю я в них по указке
Удивительной птицы моей.

Николай Заболоцкий

1.1. Основные понятия

Стену можно пробить только головой.
Все остальное — только орудия.

Лешек Кумор

Обычно говорят, что *строка* (*S*) — это последовательность символов, взятых из заранее определенного *алфавита*. При этом сразу возникают как минимум два вопроса, заключенные в понятиях «последовательность» и «алфавит».

Последовательность — это значит строка как одномерная структура, в которой каждый элемент (*символ*) имеет уникальную метку в виде номера, а рассматриваются только конечные элементы строки (первый и последний). Далее, каждый элемент строки, кроме первого (самого левого), имеет единственный предшествующий элемент, а каждый элемент, кроме последнего (самого правого), имеет единственный следующий элемент. Проводя аналогию с языком программирования Паскаль, можно сказать, что для строки работают операции `pred(`*i*`)` и `succ(`*i*`)`, где *i* — номер символа в строке, и справедливы равенства: *i* = `succ(pred(`*i*`))` (за исключением самого левого символа) и *i* = `pred(succ(`*i*`))` (за исключением самого правого символа).

Алфавит — это интуитивно достаточно ясное понятие; определим его как конечное множество *A* различимых элементов, на котором определено *отношение порядка*. Традиционные примеры алфавитов: латинский; русский; все символы в соответствии с их кодировкой ASCII. Алфавит из двух символов (а компьютер работает с данными, представленны-

ми в таком алфавите!) называют *бинарным* (*двоичным*). Отношение порядка для символов из алфавита говорит о том, что для любых $x \in A$ и $y \in A$ можно сделать вывод, какой элемент больше другого, т. е. что $x < y$ или $y < x$ при $x \neq y$.

Понятие *«подстрока» строки S* определяется как $S[i..j]$ для любой пары таких чисел i и j, что $1 \leqslant i \leqslant j \leqslant n$, где n — количество символов в S (*длина строки*, обозначим ее как $|S|$). Если же $i > j$, то мы считаем подстроку $S[i..j]$ *пустой*. Другими словами, подстрока — это часть строки, состоящая из некоторого количества смежных символов исходной строки, и в данном случае «некоторое количество» определяется как $j - i + 1$. Можно выделить точно $n - k + 1$ подстрок длины k из строки S длины n: это подстроки $S[1..k]$, $S[2..k + 1]$, ..., $S[n - k + 1..n]$. Общее количество подстрок с длинами от 1 до n определяется суммой — $\sum_{k=1}^{n}(n-k+1) = \frac{n^2+n}{2}$, т. е. имеет порядок $O(n^2)$.

Строки (естественно, на одном алфавите) S_1 и S_2 равны, если:

1) совпадают их длины (n);
2) $S_1[i] = S_2[i]$ для всех $i = 1, \ldots, n$.

На множестве строк на упорядоченном алфавите A отношение порядка (лексикографического) вводится естественным образом. Пусть имеется две строки $S_1[1..n]$ и $S_2[1..m]$, тогда мы говорим, что $S_1 < S_2$ (S_1 лексикографически меньше S_2), если выполняется одно из следующих условий (взаимоисключающих):

а) $n < m$ и $S_1[1..n] = S_2[1..n]$;

б) существует такое целое число i ($i \in 1..\min\{n, m\}$), что $S_1[1..i - 1] = S_2[1..i - 1]$ и $S_1[i] < S_2[i]$ (или, другими словами, i — первая позиция слева, в которой элементы S_1 и S_2 различны).

Примеры

`abc` < `abcd` ($n = 3$, $m = 4$); `abdef` < `ada` ($i = 2$, `b` < `d`).

Префикс строки S, заканчивающийся в позиции i, — это подстрока $S[1..i]$.

Суффикс строки S, начинающийся в позиции i, — это подстрока $S[i..n]$.