УДК 004.021

Студ. А.Н. Зайцев
Науч. рук. ст. преп. Е.А. Блинова
(кафедра информационных систем и технологий, БГТУ)

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ ДЛЯ ПОДСЧЕТА ПАЛИНДРОМОВ В СТРОКЕ

Определение палиндрома — число (например, 404), буквосочетание, слово или текст, одинаково читающееся в обоих направлениях. Иногда палиндромом называют любой симметричный относительно своей середины набор символов. Сфера применения алгоритмов для подсчёта количества палиндромов — от алгоритмов для анализа цепочек ДНК и РНК в биологии до решения математических задач на нахождение чисел Лишрел [1, 2].

В ходе работы были рассмотрены различные алгоритмы подсчета количества палиндромов, сгенерированы палиндромы различной длины и состава, проанализированы результаты выполнения алгоритмов.

Рассмотрим элементарный алгоритм: перебираются левая и правая границы текущей подстроки, после этого текущая подстрока проверяется, является ли она палиндром с помощью тривиального алгоритма (производится итерация от центра проверяемой строки, чтобы зеркальные символы были равны). Плюсы алгоритма: простота реализации и понимания, малая скрытая константа. Минусы: крайне малая скорость работы, асимптотика $O(N^3)$.

Следующий алгоритм заключается в нахождении радиуса наибольшего палиндрома с центром в данном символе методом проверки зеркальных символов по порядку, начиная от центрального. По радиусам определяем количество палиндромов. Плюсы алгоритма: легко реализуется, быстро работает на случайных данных. Минусы: на специальных данных малая скорость работы, асимптотика $O(N^2)$.

Далее рассматриваем алгоритм с использованием техники хеширования, который заключается в вычислении хеш-функции для каждого префикса и суффикса данной строки. По этим данным мы можем получить значение хеш-функции для любой подстроки за O(1). После этого перебираем символы строки, и для каждого символа находим радиус наибольшего палиндрома с центром в данном символе. Наибольший радиус находим с помощью метода дихотомии. Сравниваем строки при помощи подсчитанных ранее хешей. Плюсы: асимптотическая сложность NlogN. Минусы: сложная реализация, низкая скорость работы на случайных тестах, возможность возникновения

коллизии — хеш-функция может выдать неправильный результат (например, на строках Туэ-Морса) [3].

Алгоритм Манакера заключается в том, что поддерживая самый правый палиндром в строке, мы обрабатываем строку символ за символом. На каждой итерации проверяем, находится ли текущий элемент внутри границ самого правого палиндрома или нет. Если находится, то можем извлечь ответ из ранее посчитанных значений. Если же не находится, то продвигаемся еще на один символ и сравниваем зеркальные элементы относительно центра, обновляя при этом границы самого правого найденного палиндрома[4]. Таким образом, каждый пройденный символ строки продвигает в подсчете числа палиндромов. Плюсы: высокая скорость работы, хорошо оптимизируется по кеш-линии, самая лучшая оценка сложности O(N). Минусы: сложность реализации.

Проанализируем нахождение количества палиндромов с использованием структуры данных «дерево палиндромов» [5]. В вершинах дерева будут находиться палиндромы. Вершины между собой соединены рёбрами, которое указывает на то, какую букву нужно добавить с обеих сторон к палиндрому в данной вершине, чтобы перейти в вершину, куда ведёт ребро. Также существуют две фиктивные корневые вершины для удобной реализации алгоритма. Суффиксная ссылка будет вести в вершину, которая также является палиндромом и которая является наибольшим собственным суффиксом данной вершины. Символы строки добавляются в дерево по одному. Плюсы: решает широкий спектр задач про палиндромы, легок для понимания и реализации. Минусы: работает медленнее алгоритма Манакера.

Время выполнения алгоритмов приведено в таблице 1.

Таблица 1 - Сравнительный анализ скорости работы (i7-3630QM)

Алгоритм	Оценка сложности алгоритма	N = 100	N = 1000
Тривиальный алгоритм	$O(N^3)$	2620 msec	18441548 msec
Подсчет радиусных палиндромов	$O(N^2)$	451 msec	157004 msec
Алгоритм с использованием техники хеширования	O(NlogN)	1643 msec	320603 msec
Алгоритм Манакера	O(N)	328 msec	83343 msec
Дерево палиндромов	O(NlogN)	1006 msec	172050 msec

Также до открытия дерева палиндромов использовался алгоритм суффиксного дерева Укконена с решением на нём задачи LCA алгоритмом Фарах-Колтон-Бендера. Но в силу того, что данная структура

данных похожа на дерево палиндромов, она не рассматривается в данной статье.

Тест показал, что не всегда только асимптотические оценки важны при выборе алгоритма для решения задачи. У каждого алгоритма в асимптотике есть скрытая константа. Она зависит от количества операций в алгоритме, сложности операций, способность компилятора кешировать некоторые промежуточные шаги алгоритма. Некоторые алгоритмы можно выполнять на нескольких потоках, что даст ощутимый выигрыш в скорости.

Стоит также заметить, что в зависимости от длины строк должен и выбираться подходящий алгоритм. На коротких строках больше выигрыш от оптимизации скрытых констант, на больших же строках важнее асимптотическая оптимизация. Немаловажную роль играет и тот факт, какого вида строки. Например, если в строках мало палиндромов, то алгоритм с O(N2) сравняется с алгоритмом Манакера.

Генерация тестов производилась следующим образом: задавалась максимальная длина строки и количество строк. После этого случайно выбирались размер алфавита (использовались все строчные латинские буквы) и длина строки. Далее строка заполнялась случайными символами из выбранного алфавита.

С моей реализацией данных алгоритмов можно ознакомиться по ссылке: https://github.com/ZaMaZaN4iK/AlgoForPalindromes

ЛИТЕРАТУРА

1. Palindromic sequence [Электронный ресурс] / Wikipedia. – 2016. / Режим доступа: https://en.wikipedia.org/wiki/Palindromic_sequence. – Дата доступа: 28.04.2016.

2. Lychrel number [Электронный ресурс] / Wikipedia. – 2016. / Режим доступа: https://en.wikipedia.org/wiki/Lychrel_number. – Дата доступа: 28.04.2016.

3. Thue–Morse sequence [Электронный ресурс] / Wikipedia. – 2016. / Режим доступа: 4https://en.wikipedia.org/wiki/Thue%E2%80%93Morse_sequence. – Дата доступа: 28.04.2016.

4. Нахождение всех подпалиндромов [Электронный ресурс] / E-MAXX. – 2016. / Режим доступа: 4http://e-maxx.ru/algo/palindromes_count. – Дата доступа: 28.04.2016.

5. Palindromic Tree [Электронный ресурс] / ADILET.org. – 2016. / Режим доступа: http://adilet.org/blog/25-09-14/. – Дата доступа: 28.04.2016.