Лабораторная работа №8 по курсу дискретного анализа: Динамическое программирование

Выполнил студент группы М8О-312Б-22 МАИ *Мамонтов Егор*.

Условие

Вариант: 6

Палиндромы

Задана строка S состоящая из n прописных букв латинского алфавита. Вычеркиванием из этой строки некоторых символов можно получить другую строку, которая будет являться палиндромом. Требуется найти количество способов вычеркивания из данного слова некоторого (возможно, пустого) набора таких символов, что полученная в результате строка будет являться палиндромом. Способы, отличающиееся только порядком вычеркивания символов, считаются одинаковыми.

Метод решения

Для решения этой задачи я использовал динамическое программирование. Основная идея заключается в построении двумерного массива, где каждый элемент [i][j] хранит количество способов получить палиндром из подстроки input_str[i...j]. Сначала я прошёлся по базовым случаям для длины подстрок 1 и 2 и далее использовал метод динамического программирования для последующих длин подстрок, сравнивая крайние символы.

Описание программы

Для реализации алгоритма были реализованы следующие функции и структуры:

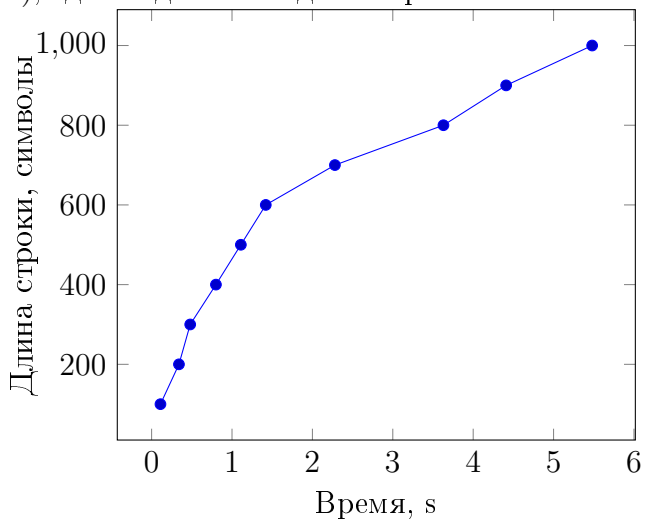
- `void baseSollutions` - функция, которая проходится по базавым случаям. Если подстрока из 1 символа, то она палиндром. Если два соседних символа одинаковые, то палиндромов 3, иначе 2. Для подстрок длиной 3 уже много писанины, поэтому далее идем за помощью к динамическому програмированию.
- `void dynamicProg` - функция, которая сранивает подстроки, длиной более двух символов. Он учитывает кол-во палиндромов во внутренних подстроках и находит кол-во палиндромов в подстроке.

Дневник отладки

1. Был получен WA на тесте №17. Для решения проблемы нужно было в dynamicProg функции идти от длины подстрок более 2.
2. Был получен WA на тесте №1. Опечатался.
3. Был получен OK.

Тест производительности

Сложность алгоритма зависит от сложности заполнения массива, то есть сложность $O(n^2)$, где n - длина входной строки.



Выводы

Я научился лучше пользоваться динамическим програмированием, учитывая прошлые результаты. Задачу решил с использованием двумерного массива для хранения промежуточных результатов. Это позволило эффективно вычислить количество палиндромов за $O(n^2)$.