Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №7 по курсу «Дискретный анализ»

Студент: И. Д. Черненко
Преподаватель: А. А. Кухтичев
Группа: М8О-306Б

Дата:
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №7

Задача: При помощи метода динамического программирования разработать алгоритм решения задачи, определяемой своим вариантом; оценить время выполнения алгоритма и объем затрачиваемой оперативной памяти. Перед выполнением задания необходимо обосновать применимость метода динамического программирования.

Разработать программу на языке C или C++, реализующую построенный алгоритм. Формат входных и выходных данных описан в варианте задания:

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.

Формат входных данных

Задана одна строка $\mathbf{S(|S| \leq 100)}$.

Формат результата

Необходимо вывести одно число – ответ на задачу. Гарантируется, что он $\leq 2^{63}$ **- 1.**

1 Описание

Ключевая идея в динамическом программировании достаточно проста. Как правило, чтобы решить поставленную задачу, требуется решить отдельные части задачи (подзадачи), после чего объединить решения подзадач в одно общее решение. Часто многие из этих подзадач одинаковы. Подход динамического программирования состоит в том, чтобы решить каждую подзадачу только один раз, сократив тем самым количество вычислений. Это особенно полезно в случаях, когда число повторяющихся подзадач экспоненциально велико. В данной задаче мы введем двумерный массив D и в каждой ячейке D_{ij} будем хранить количество способов получить палиндром в строке s с i-ой по j-ю позицию. Так, если s[i] = s[j], то:

- считаем значение для D[i+1][j], D[i][j-1] и прибавляем единицу т.к. решение получается из равенства символов;

Если же символы не равны:

- считаем значение для D[i][j-1], D[i+1][j] и вычитаем решение из середины D[i+1][j-1];

Идея такого решения появилась из классической задачи динамического программирования: поиска наибольшей общей подпослдеовательности.

2 Исходный код

```
#include <iostream>
#include <string>
typedef unsigned long long ull;
ull matrix[101][101];
std::string string;
ull DoCount(int i, int j) {
    if (i > j) {
        return 0;
    }
    if (i == j) {
        return 1;
    }
    if (matrix[i][j] == 0) {
         if (string[i] == string[j]) {
             matrix[i][j] = DoCount(i + 1, j) + DoCount(i, j - 1) + 1;
         } else {
             matrix[i][j] = DoCount(i + 1, j) + DoCount(i, j - 1) - DoCount(i + 1, j -
                 1);
         }
         return matrix[i][j];
    } else {
        return matrix[i][j];
    }
}
 int main() {
    std::cin >> string;
    std::cout << DoCount(0, string.length() - 1) << std::endl;
    return 0;
}
```

3 Консоль

```
zebr@DESKTOP-T2MUUB0:/mnt/d/Study/DA_lab7/cmake-build-debug$ ./DA_lab7
38943432983521435346436
354353254328383
+
90409438473849324729384738 43
2343543
-
972323
2173937
>
2
3
-
3894343333787468967 4819
90409438473849324729361303 00
false
Error
```

4 Выводы

Выполнив седьмую лабораторную работу по курсу «Дискретный анализ» я познакомился с идеями динамического программирования. Динамическое программирование оказывается очень полезным в задачах оптимизации, такие задачи имеют много возможных решений, ищется наиболее оптимальное из них.Кроме того алгоритмы динамического программирования позволяют найти решение тогда, когда наивный алгоритм вообще не сможет дать ответа за адекватное время.

Список литературы

[1] *Палиндром - C++ для приматов*
URL: `https://cpp.mazurok.com/tag/палиндром/` (дата обращения: 11.03.2021).

[2] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. *Алгоритмы: построение и анализ, 2-е издание.* — Издательский дом «Вильямс», 2013. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. — 256 с. (ISBN 5-8459-0857-4 (рус.)) (дата обращения: 11.02.2021).