Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №4 по курсу «Дискретный анализ»

Студент: П. Ф. Гришин
Преподаватель: С. А. Михайлова
Группа: М8О-201Б-21

Дата:
Оценка:
Подпись:

Москва, 2023

Лабораторная работа №1

Задача: Необходимо реализовать поиск одного образца в тексте с использованием алгоритма Z-блоков. Алфавит — строчные латинские буквы.

Формат ввода: На первой строке входного файла текст, на следующей — образец. Образец и текст помещаются в оперативной памяти.

Формат вывода: В выходной файл нужно вывести информацию о всех позициях текста, начиная с которых встретились вхождения образца. Выводить следует по одной позиции на строчке, нумерация позиций в тексте начинается с 0.

1 Описание

В задаче поика подстрок у нас есть две строк: текстовая строка T и шаблонная строка P. Мы хотим найти все вхождения P в T. Для этого мы будем использовать Z-функцию $z[i]$. Она представляет из себя массив длинной строки T и каждый его элемент представляет из себя наибольший общий префикс этой строки на i-ой позиции. Однако, чтобы найти подстроки нам понадобится строка длины $n+m+1$, где n и m - это длинны текста T и шаблона P соответственно. Этот массив будет на единицу больше, т.к. итоговая строка, которую будет принимать Z-функция, выглядит так: $P\#T$, где $\#$ - это специальный символ, который будет останавливать Z-функцию на дальнейщую проверку символов.

2 Исходный код

Теперь поговорим непосредственно над реализацией Z-блоков.

Z-блоком назовем подстроку с началом в позиции i и длиной $Z[i]$. Для работы алгоритма заведём две переменные: $left$ и $right$ - начало и конец Z-блока строки S с максимальной позицией конца $right$ (среди всех таких Z-блоков, если их несколько, выбирается наибольший). Изначально $left = 0$ и $right = 0$. Пусть нам известны значения Z-функции от 0 до $i - 1$. Найдём $Z[i]$. Рассмотрим два случая:

- $i > right$: Просто пробегаемся по строке S и сравниваем символы на позициях $S[i+j]$ и $S[j]$.Пусть j первая позиция в строке S, для которой не выполняется равенство $S[i + j] = S[j]$, тогда j это и Z-функция для позиции i. Тогда $left = i$,$right = i + j - 1$. В данном случае будет определено корректное значение $Z[i]$ в силу того, что оно определяется наивно, путем сравнения с начальными символами строки.

- $i \leq right$: Сравним $Z[i - left] + i$ и $right$. Если $right$ меньше, то надо просто наивно пробежаться по строке начиная с позиции $right$ и вычислить значение $Z[i]$. Корректность в таком случае также гарантирована.Иначе мы уже знаем верное значение $Z[i]$, так как оно равно значению $Z[i - left]$.

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  vector<int> zFunction(string& s) {
7      int n = s.length();
8      vector<int> z(n);
9      z[0] = n;
10     for (int i = 1, l = 0, r = 0; i < n; ++i) {
11         if (i <= r) {
12             z[i] = min(r - i + 1, z[i - l]);
13         }
14         while (i + z[i] < n && s[z[i]] == s[i + z[i]]) {
15             ++z[i];
16         }
17         if (i + z[i] - 1 > r) {
18             l = i;
19             r = i + z[i] - 1;
20         }
21     }
22     return z;
23 }
24
25 int main() {
```

```
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    string strForFunc, pattern;
    cin >> strForFunc;
    cin >> pattern;
    string finalString = pattern + "#" + strForFunc;
    vector<int> zV = zFunction(finalString);
    int patternSize = pattern.length();
    for (int i = patternSize; i < zV.size(); i++) {
        if (zV[i] == patternSize) cout << i - 1 - patternSize << "\n";
    }
}
```

3 Консоль

```
gpavel@gpavel-HP-Pavilion-Gaming-Laptop-17-cd1xxx:~/Desktop/DA/Lab4$ c++ main.cpp
gpavel@gpavel-HP-Pavilion-Gaming-Laptop-17-cd1xxx:~/Desktop/DA/Lab4$ ./a.out
abcdaattta
aatt
4
gpavel@gpavel-HP-Pavilion-Gaming-Laptop-17-cd1xxx:~/Desktop/DA/Lab4$ cat test.txt
uqjuuwnyftuxfztqvvrcurlxwuhaxsubfmtapbtwafvxgnhvbnlrcvswthruybiukoqpbmducirvoxginpqiyb
nlrcvswthruybiukoqpbmducirv
gpavel@gpavel-HP-Pavilion-Gaming-Laptop-17-cd1xxx:~/Desktop/DA/Lab4$ ./a.out
<test.txt
49
```

4 Тесты

Так как проверяется всего один раз, то была сгенерирована довольно большая текстовая строка и паттерн длиной 20. Сравнение будет производится со встроенной функцией std::find в C++.

```
#100000
gpavel@gpavel-HP-Pavilion-Gaming-Laptop-17-cd1xxx:~/Desktop/DA/Lab4$ ./a.out
<pushTest.txt
100
Time taken by function: 141 microseconds
gpavel@gpavel-HP-Pavilion-Gaming-Laptop-17-cd1xxx:~/Desktop/DA/Lab4$ c++ main.cpp
gpavel@gpavel-HP-Pavilion-Gaming-Laptop-17-cd1xxx:~/Desktop/DA/Lab4$ ./a.out
<pushTest.txt
100
Time taken by function: 11 microseconds
```

Всего была найдена одна строка из-за того, что в строке довольно много уникальных комбинаций, а также длина полученного паттерна довольно большая для поиска.

Переидем к анализу полученных данных. Функция find работает в 12 раз быстрее, чем написанный мною алгоритм. Хоть у обоих алгоритмов временная сложность одна и та же. Осмелюсь предположить, что такая разница связана с тем, что позиция сразу выводилась при использовании find, а Z-функция требует просмотра массива. Однако такие изменения незначительно повлияют на время работы, т.к. доступ к элементу массива произойден мгновенно.

И другая причина плохой работы программы - неграмотная реализация поиска подстроки в строке. Возможно, можно было бы придумать другой способ быстрого поиска.

5 Выводы

Выполнив четвёртую лабораторную работу по курсу «Дискретный анализ», я вспомнил уже знакомые мне и узнал новые алгоритмы подиска подстроки в строке, а также научился реализовывать алгоритм Z-блока. При тестировании понял, что программа работает медленно и требует дальнейших наработок или другого подхода к решению, иначе, использование в реальных больших проектах этого алгоритма приведет к довольно долгой обработке данных.

Список литературы

[1] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. *Алгоритмы: построение и анализ, 2-е издание.* — Издательский дом «Вильямс», 2007. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. — 1296 с. (ISBN 5-8459-0857-4 (рус.))