

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: А. В. Барсов
Преподаватель: Н. С. Капралов
Группа: М8О-307Б-19
Дата:
Оценка:
Подпись:

Москва, 2022

Лабораторная работа №4

Задача: Необходимо реализовать один из стандартных алгоритмов поиска образцов для указанного алфавита.

Запрещается реализовывать алгоритмы на алфавитах меньшей размерности, чем указано в задании.

Вариант алгоритма: Поиск одного образца при помощи алгоритма Кнута-МоррисаПратта.

Вариант алфавита: Слова не более 16 знаков латинского алфавита (регистроне-зависимые)

1 Описание

Требуется реализовать алгоритм Кнута-Морриса-Пратта для поиска подстроки в строке. Учитывая, что алфавит состоит из регистронезависимых слов не более 16 знаков, нужно уметь правильно представлять переводы строки, пробелы и табуляции.

Согласно [1], алгоритм Кнута-Морриса-Пратта прикладывает образец к тексту и начинает делать сравнение с левого конца. В случае полного совпадения, было найдено вхождение, сдвигаем образец на один символ вправо. Если же есть несовпадения, то мы делаем сдвиг по особому правилу, в отличие от алгоритма наивного поиска, который всегда сдвигает на один символ.

Для каждой позиции i определим $sp_i(P)$ как длину наибольшего собственного суффикса $P[1..i]$, который совпадает с префиксом P , при чём символы в позициях $i + 1$ и $sp_i(P) + 1$ различны.

Для каждой позиции i , определим $Z_i(P)$ как длину префикса строки $P[i..|P|]$, который совпадает с префиксом P . Причём $Z_0(P)$ принято считать равным 0. Набор таких значений называется Z -функцией строки P . Z -функция является известным алгоритмом и может быть вычислена за линейное время от длины строки, как описано в [3].

Значения $Z_j(P)$ соответствует такому $sp_i(P)$, что $i = j + Z_j(P) - 1$. Таким образом вычисление всех $sp_i(P)$ имеет сложность $O(n)$, где n — длина образца.

Будем делать сдвиг, используя вычисленное в каждой позиции значение $sp_i(P)$. Если при сравнении было найдено несовпадение в позиции $i+1$, то мы можем сделать сдвиг на $i - sp_i(P)$, не теряя вхождений.

Алгоритм Кнута-Морриса-Пратта сравнивает каждый символ не более двух раз, то есть совершает не более $2 * m$ сравнений символов, где m — длина текста. Учитывая сложность препроцессинга, асимптотика алгоритма $O(n + m)$.

2 Консоль

```
[artem@IdeaPad solution]$ make
g++ -pedantic -Wall -std=c++11 -Werror -Wno-sign-compare -O2 -lm -c main.cpp
g++ -pedantic -Wall -std=c++11 -Werror -Wno-sign-compare -O2 -lm -c search.cpp
g++ -pedantic -Wall -std=c++11 -Werror -Wno-sign-compare -O2 -lm main.o search.o
-o solution
[artem@IdeaPad solution]$ ./solution
cat dog cat dog bird
CAT dog CaT Dog Cat DOG bird CAT
dog cat dog bird
1,3
1,8
```

3 Выводы

Во время выполнения лабораторной работы я вспомнил основные понятия, связанные со строками, изучил алгоритм Кнута-Морриса-Пратта и его варианты с препоцессингом через Z-функцию и префикс-функцию строки.

Задачи поиска часто встречаются в жизни, будь то поиск забытых вещей или нужной книги в библиотеке. Конкретно алгоритмы поиска подстроки используются в текстовых редакторах и браузере.

Я дополнительно ознакомился с другими алгоритмами поиска образцов в тексте. Для чего нужно так много алгоритмов? Каждый алгоритм хорош в отдельном случае. Например алгоритм Кнута-Морриса-Пратта хорошо ищет один образец в большом тексте, но не может эффективно найти несколько образцов в не самом большом тексте, а алгоритм Ахо-Корасик наоборот больше подходит для поиска множества образцов.

Список литературы

- [1] Гасфилд Дэн. *Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология*. Издательский дом «БХВ-Петербург». Перевод с английского: И. В. Романовский. — 654 с. (ISBN 5-7490-0103-8 ("БХВПетербург"))
- [2] Справочник по алгоритмам и структурам данных. *Префикс-функция. Алгоритм Кнута-Морриса-Пракса*.
URL: https://e-maxx.ru/algo/prefix_function .
- [3] Справочник по алгоритмам и структурам данных. *Z-функция строки и её вычисление*.
URL: https://e-maxx.ru/algo/z_function .