Лабораторная работа № 4 по курсу дискретного анализа: Строковые алгоритмы

Выполнил студент группы 08-208 МАИ *Ибрагимов Далгат.*

Условие

Общая постановка задачи: 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.

Метод решения

Задача состоит в том, чтобы найти все вхождения шаблона в тексте, используя алгоритм Z-блоков. Этот алгоритм эффективен для поиска подстроки в строке и имеет линейное время работы O(n+m), где n - длина текста, а m - длина шаблона.

Шаги решения:

- Построение Z-функции: Сначала мы строим Z-массив для строки, которая представляет собой конкатенацию шаблона и текста, разделённых специальным символом (например, "$").
- Поиск вхождений: В Z-массиве все элементы, равные длине шаблона, указывают на начало вхождения шаблона в текст.

Описание программы

Основными алгоритмами, структурами и функциями стали:

- строка concat: Мы конкатенируем шаблон и текст через специальный символ $, чтобы избежать совпадений между концом шаблона и началом текста.
- std::vector<int> computeZFunction(const std::string s) - эта функция строит Z-функцию, которая для каждой позиции i строки concat определяет длину максимальной подстроки, которая начинается в i и совпадает с префиксом строки.
- поиск вхождений: Если значение z[i] равно длине шаблона, это означает, что в позиции i - patternLength - 1 в тексте начинается вхождение шаблона.

Дневник отладки

Основной проблемой был выбор неправильного метода компиляции в контесте

Тест производительности

```
lockr@lockR:~/projects/DA_LABS/lab4/test$ g++ test.cpp
lockr@lockR:~/projects/DA_LABS/lab4/test$ ./a.out
Z-algorithm took 0.000499741 seconds.
std::string::find took 0.000128996 seconds.
```

Z-блоки (Z-algorithm): Этот алгоритм оптимален для поиска подстрок, особенно когда необходимо найти все вхождения шаблона в тексте. Время выполнения линейное относительно длины текста и шаблона. std::string::find: Этот метод стандартной библиотеки часто работает быстрее на коротких строках из-за оптимизаций, но на больших строках может проигрывать специализированным алгоритмам, таким как Z-блоки.

Выводы

В результате данной лабораторной работы была написана и отлажена программа на языке C++, реализующая поиск одного образца при помощи алгоритма Z-блоков. Я реализовала алгоритм КМП с использованием сильной префикс-функции и Z-функции. реальных проектах и задачах выбор алгоритма должен зависеть от конкретных требований: размер данных, количество подстрок, которые необходимо найти, и частота вызова функции поиска. Если требуется простой и быстрый поиск, стоит использовать std::string::find. В более сложных случаях, когда производительность критична и необходимо обрабатывать большие объемы данных, можно рассмотреть Z-алгоритм.