Лабораторная работа № 4 по курсу дискретного анализа: Исследование качества программ

Выполнил студент группы 8О-207Б-17 МАИ *Ваньков Денис.*
Задание: Необходимо реализовать один из стандартных алгоритмов поиска образцов для указанного алфавита.

Условие

1. Поиск одного образца в тексте и спользованием алгоритма Апостолико - Джанкарло. Алфавит - строчные латинские буквы. На первой строке входного файла подается образец, а на следующей - текст. Образец и текст помещается в оперативной памяти. 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.

2. Вариант задания 3-1. Тип алгоритма: Поиск одного образца при помощи алогоритма Апостолико-Джанкарло. Тип слова: Слова не более 16 знаков латинского алфавита (регистронезависимые).

Метод решения

Алгоритм Апостолико - Джанкарло основывается на алгоритме Боера - Мура, однако имеет улучшения. Данный алгоритм сохраняет все преимущетсва сдвигов в алгоритме Боэра - Мура, и допускает простой анализ времени счета с линейной оценкой для наихудшего случая.

Во - первых, мы делим работу алгоритма Боэра - Мура на фазы сдвига и сдвига. В кажой фазе правый конец Образца прикладывается к какому-то символу из Текста, и образец сравнивается справа налево с выбранными символами из Текста до обнаружения либо не совпадения в каком - то символе, либо полного совпадения. После этого Образец сдвигается вправо на расстояние, согласно правилам Боэра - Мура.

Во - вторых, будем считать также, что вектор N получен в препроцессинге Образца. Далее сделаем 2 модификации:

1). После самого препроцессинга будем работать с вектором M, с длиной m, и на каждой фазе модифицировать не более одного элемента.

2). Вторая модификация использует векторы M и N и ускоряет алгоритм Боэра - Мура за счет пропуска некоторых проверок.

Описание программы

Я описал свою реализацию вектора, которая содержится в файле *vector.h*, а также реализовал очередь для хранения текста в файле *queue.h*.

В файле основной программы имеем 2 функции препроцессинга для алгоритма Боэра - Мура, для правила плохого символа и правила хорошего суффикса. Также дополнительную функцию определяющую максимальный сдвиг, выбирающую его из двух предыдущих функций. А также основная функция алгоритма Апостолико - Джанкарло, выполняющую непосредсвенно проверку на вхождение Образца в Текст, а также выводящюю результаты.

На вход программе подается образец, который хранится в векторе,описанном выше, затем запускается функция алгоритма Апостолико - Джанкарло с этим образцом. Внутри этой функции происходит препроцессинг для данного образца, а также считывается сам Текст, в котором будет происходить поиск этого образца. Если в результате проверки вхождения обнаружилось несовпадение, то алгоритм сдвигает (на самом деле просто меняет индекс) на максимальную длину, вычисляемую из функций препроцессинга и фаза поиска наичнается заного. В случае нахождения вхождения Образца в Тексте мы выводим номер строки текста и номер слова с которого нашлось вхождение, ксли таких входений несколько выводятся все. Затем после этого, Образец сдвигается по правилам Боэра - Мура.

Дневник отладки

В процессе отладке удалось отследить 1 утечку памяти и исправить ёё, а также пофиксить различные баги, всплывающие по ходу написания программы. Исправленная программа выполняется за то же время, однако в исправленной происходит высвобождение памяти.

Выводы

Данный алгоритм корректно находит все вхождения Образца в Текст и работает за линейное время даже в самом худшем случае. Выполняет сравнение не более 2m раз, где m длина текста, и выполняет не более O(m) дополнительных действий. Это происходит из - за того, что у нас число совпадений не более чем число символов в Тексте, также как и число не совпадний, так как проверенный символ мы не больше встречаем. В любом случае алгоритма проверит 1 раз символ, если он попал в наш блок проверки, а если не попал тоже за константное значение, так как эти блоки не пересекаются.