

# Лабораторная работа № 4 по курсу дискретного анализа: Поиск образца в строке

Выполнил студент группы 08-207 МАИ *Лебедев Иван*.

## Условие

1. Необходимо реализовать один из стандартных алгоритмов поиска образцов для указанного алфавита.
2. Поиск одного образца-маски: в образце может встречаться «джокер», равный любому другому символу. При реализации следует разбить образец на несколько, не содержащих «джокеров», найти все вхождения при помощи алгоритма Ахо-Корасик и проверить их относительное месторасположение.

## Метод решения

В первой строке считывается образец, делится по джокерам на подобразцы, и по ним строится trie. После добавления всех подобразцов в trie происходит его прошивка. Затем считывается весь текст и в нём производится поиск. При поиске создается вектор равный размеру текста и заполняется нулями. При нахождении любого образца в этом векторе элемент под индексом начала подобразца в тексте увеличивался на единицу. Весь образец считается найденным если вектор[i] равен количеству подобразцов.

## Описание программы

1. *Заголовочные файлы:*  
TNode.h(Объявление узла и функций)
2. *Cpp файлы:*  
TNode.cpp(Описание функций объявленных в TNode.h)  
main.cpp(Считывание данных, управление программой)

## Дневник отладки

Программа не проходила тест №3 из-за ошибки при подсчете индексов, и как следствие выход за границы массива. Так же была ошибка при нумерации строк(не считались пустые строки).

## Выводы

С помощью алгоритма Ахо-Корасика можно искать сразу несколько образцов в строке за линейную сложность. Благодаря этому алгоритм обрел большую популярность среди программистов. Время работы также зависит от организации данных. Если таблицу переходов в дереве хранить как индексный массив — увеличивается расход памяти, но уменьшается вычислительная сложность. Если таблицу переходов хранить как красно-чёрное дерево — расход памяти снижается, однако вычислительная сложность увеличивается.