

# Лабораторная работа № 5 по курсу дискретного анализа: суффиксные деревья

Выполнил студент группы 08-308 МАИ *Шарапов Леонид*.

## Условие

1. Общая постановка задачи.

Необходимо реализовать алгоритм Укконена построения суффиксного дерева за линейное время. Используются строчные буквы латинского алфавита.

2. Вариант 4.

Линеаризовать циклическую строку, то есть найти минимальный в лексикографическом смысле разрез циклической строки.

## Метод решения

1. Считать строку
2. Удвоить строку, так как это нужно для варианта задания
3. Составить суффиксное дерево с помощью алгоритма Укконена
4. Начиная с корня, спускаться по самым младшим дочерним узлам, количество распечатанных символов меньше длины строки (не равно)
5. Распечатать строки, соответствующие им

## Описание программы

Весь код хранится в `main.cpp`

Используемые типы данных

1. `const int`
2. `class suffixTree` - каждый его объект соответствует состоянию класса
3. `int`
4. `int*`
5. `std::string`
6. `std::shared_ptr` - для автоматического очищения памяти

7. struct active - хранит состояние, позицию символа и длину подстроки активной точки

Используемые функции

- (a) suffixTree::suffixTree(int start, int\* end) - конструктор, заполняющий состояние. Дочерние указатели зануляются, суффиксная ссылка ведет на корень, а начальная и конечная позиции в тексте суффикса задаются по переданным параметрам
- (b) suffixTree::suffixLength() - возвращает длину суффикса
- (c) suffixTree::addCharacterToTree(int pos) - добавляет в суффиксное дерево символ, стоящий на позиции pos в тексте
- (d) suffixTree::linearizeTheCyclicString() - выводит минимальный лексикографический разрез заданной строки
- (e) suffixTree::setSuffixLink() - устанавливает суффиксную ссылку для состояния на корень

## Дневник отладки

1. wrong answer test01.t - в выводе не было перехода на новую строку
2. runtime error test04.t got signal 6 - неправильное удлинение строк, то есть не было нормальной перестановки указателя со старой памяти на новую и очистки старой памяти, поэтому был выход за границу массива
3. runtime error test09.t got signal 11

## Выводы

С помощью алгоритма Укконена можно быстро построить суффиксное дерево, применяемое для поиска образца в строке, быстрое нахождение последних символов циклических сдвигов строки в преобразовании Барроуза-Веллера, задача о словаре и другие приложения. Очень сложная теория. Основные проблемы были с ней и неправильной работой со строками.