Разработка алгоритма MaxPattern для поиска максимальной подстроки, которая встречается в строке не менее двух раз без пересечений, с использованием дерева суффиксов.

Решение задачи:

Построено дерево суффиксов для входной строки.

Разработаны методы для заполнения индексов суффиксов, минимальных и максимальных индексов поддерева, и максимальной длины непересекающихся меток.

Реализован метод для поиска узла с максимальным значением maxLen.

Получена максимальная подстрока на основе найденного узла.

Результаты:

Задача была решена полностью. Алгоритм успешно находит максимальную подстроку, которая встречается не менее двух раз без пересечений.

Сложность алгоритма:

Вычислительная сложность построения дерева суффиксов составляет O(n), где n - длина строки. Заполнение индексов и вычисление maxLen также выполняются за O(n), что делает общее время выполнения алгоритма линейным.

Решение для двух файлов:

Для поиска общей подстроки в двух разных файлах можно объединить строки из двух файлов, добавив уникальный разделитель между ними. Построив дерево суффиксов для объединенной строки, можно найти общие подстроки, которые присутствуют в обоих файлах. При этом необходимо учитывать разделитель, чтобы убедиться, что найденные подстроки действительно присутствуют в обоих исходных файлах.