Занятие 3. Поиск образца в тексте.

Цель. Получить знания и навыки применения алгоритмов поиска в тексте подстроки (образца).

Задание

Для каждой задачи варианта:

1. Выполнить разработку программы, выполняя все этапы разработки.
2. Включить в этап «Описание модели (подход к решению)» описание алгоритма рассматриваемого метода. Разобрать алгоритм на примере. Подсчитать количество сравнений для успешного поиска первого вхождения образца в текст и безуспешного поиска. Определить функцию (или несколько функций) для реализации алгоритма. Определить предусловие и постусловие.
3. Сформировать таблицу тестов с указанием успешного и неуспешного поиска, используя большие и небольшие по объему текст и образец и включить ее в этап тестирование.
4. Разработать программу
5. Оценить практическую сложность алгоритма в зависимости от длины текста и длины образца и отобразить результаты в таблицу (для отчета).
6. Сравнить время поиска различных методов на одних и тех же наборах данных.

Варианты

Номер теста	Задачи варианта
1	1) Линейный поиск первого вхождения подстроки в строку. 2) Используя алгоритм Бойера-Мура-Хорспула, найти последнее вхождение подстроки в строку.
2	1) Дано предложение, состоящее из слов. Сформировать массив слов – целых чисел. Словом считаем подстроку, ограниченную с двух сторон пробелами. 2) Найти все вхождения подстроки в строку, используя алгоритм Бойера-Мура (только эвристика хорошего суффикса).
3	1) Дано предложение, состоящее из слов. Найти самое длинное слово предложения, первая и последняя буквы которого одинаковы. 2) Используя алгоритм Кнута-Мориса-Пратта, найти индекс последнего вхождения образца в текст.

4	1) Дано предложение, состоящее из слов, разделенных знаками препинания. Определить, сколько раз в предложение входит первое слово. 2) Проверка на плагиат. Используя алгоритм Рабина-Карпа, проверить, входит ли подстрока проверяемого текста в другой текст.
5	1) Дано предложение, состоящее из слов, разделенных одним пробелом, удалить из него слова, встретившиеся более одного раза. 2) Дано предложение, состоящее из слов, разделенных одним пробелом. Удалить из предложения все вхождения заданного слова, применяя для поиска слова в тексте метод Кнута-Мориса-Пратта.
6	1) Дан произвольный текст, состоящий из слов, разделенных знаками препинания. Отредактировать его, оставив между словами по одному пробелу, а между предложениями по два. 2) Дана непустая строка S, длина которой N не превышает 10^6. Считать, что элементы строки нумеруются от 1 до N. Требуется для всех i от 1 до N вычислить $\pi[i]$ – префикс функцию.
7	1) Дано предложение, состоящее из слов, разделенных знаками препинания. Определить количество слов равных последнему слову, больших последнего слова. 2) Строка S была записана много раз подряд, после чего из получившейся строки взяли произвольную часть строки - подстроку и передали как входные данные. Необходимо определить минимально возможную длину исходной строки S. Реализация алгоритмом Кнута-Мориса-Пратта.
8	1) Дано предложение, слова в котором разделены пробелами и запятыми. Распечатать те слова, которые являются обращениями других слов в этом предложении. 2) Даны две строки a и b. Требуется найти максимальную длину префикса строки a, который входит как подстрока в строку b. При этом считать, что пустая строка является подстрокой любой строки. Реализация алгоритмом Кнута-Мориса-Пратта.
9	1) Дано предложение, слова в котором разделены пробелами и запятыми. Распечатать те пары слов, расстояние между которыми наименьшее. Расстояние – это количество позиций, в которых слова различаются. Например, МАМА и ПАПА, МЫШКА и КОШКА расстояние этих пар равно двум.

	2) Найти все вхождения подстроки в строку, используя алгоритм Бойера-Мура с турбосдвигом.
10	1) Дано предложение, разделенных знаками препинания. Удалить из предложения все слова, равные заданному слову. 2) Назовем строку палиндромом, если она одинаково читается слева направо и справа налево. Примеры палиндромов: "abcba", "55", "q", "xyzzyx". Требуется для заданной строки найти максимальную по длине ее подстроку, являющуюся палиндромом. Реализация алгоритмом Кнута-Мориса-Пратта.
11	1) Дан текст, состоящий из слов, разделенных знаками препинания. Сформировать массив из слов, которые содержат заданную подстроку. 2) Назовем строку палиндромом, если она одинаково читается слева направо и справа налево. Примеры палиндромов: "abcba", "55", "q", "xyzzyx". Требуется для заданной строки найти максимальную по длине ее подстроку, являющуюся палиндромом. Реализация алгоритмом Бойера-Мура-Хорспула.
12	1) Дан текст, разделенных знаками препинания. Сформировать массив из слов, в которых заданная подстрока размещается с первой позиции. 2) В текстовом файле хранятся входные данные: на первой сроке – подстрока (образец)длиной не более 17 символов для поиска в тексте; со второй строки – текст (строка), в котором осуществляется поиск образца. **Строка**, в которой надо искать, **не ограниченна по длине.** Применяя алгоритм Бойера-Мура с турбосдвигом вывести индексы строки, на которые смещается алгоритм при поиске вхождения образца.
13	1) Дан текст, состоящий из слов, разделенных знаками препинания. Сформировать массив из слов, в которых заданная подстрока размещается в конце слова. 2) В текстовом файле хранятся входные данные: на первой сроке – подстрока (образец)длиной не более 17 символов для поиска в тексте; со второй строки – текст (строка), в котором осуществляется поиск образца. **Строка**, в которой надо искать, **не ограниченна по длине.** Применяя алгоритм Рабина-Карпа определить количество вхождений в текст заданного образца.

14	1) Дан текст, состоящий из слов, разделенных знаками препинания. Переставить первое и последнее слово в тексте. 2) Дан текст и множество подстрок образцов. Определить сколько раз каждый из образцов входит в исходный текст. Реализовать на алгоритме Рабина-Карпа. Примечание: для всех образцов создать хеш-таблицу.
15	1) Дан массив ключевых слов языка C++. Упорядочить их, располагая слова в алфавитном порядке, используя обменную сортировку. 2) Дан текст и множество подстрок образцов. Определить сколько раз каждый из образцов входит в исходный текст. Реализовать алгоритм Бойера-Мура-Хорспула. Примечание. Для всех образцов создать хеш-таблицу.

Контрольные вопросы

1. Что называют, строкой?
2. Что называют префиксом строки?
3. Что называют суффиксом строки?
4. Асимптотическая сложность последовательного поиска подстроки в строке?
5. В чем особенность поиска образца алгоритмом Бойера –Мура?
6. Приведите асимптотическую сложность алгоритма Бойера –Мура поиска подстроки в строке по времени и памяти.
7. Приведите пример входных данных для реализации эффективного метода прямого поиска подстроки в строке.
8. Приведите пример строки, для которой поиск подстроки "aaabaaa" будет более эффективным, если делать его методом Кнута, Морриса и Пратта, чем, если делать его методом Бойера и Мура. И наоборот.
9. Объясните, как влияет размер таблицы кодов в алгоритме Бойера и Мура на скорость поиска.
10. За счет чего в алгоритме Бойера и Мура поиск оптимален в большинстве случаев?
11. Поясните влияние префикс-функции в алгоритме Кнута, Морриса и Пратта (КМП) на организацию поиска подстроки в строке.
12. Приведите пример префикс-функции для поиска образца в тексте для алгоритма КМП.
13. В чем особенность поиска образца алгоритмом Рабина и Карпа?
14. Приведите асимптотическую сложность алгоритма Рабина и Карпа поиска подстроки в строке.

15. Что такое бор?
16. Какие структуры хранения данных используются для реализации простого бора?
17. Приведите пример бора и реализуйте его одним из способов. Объясните алгоритм поиска образца с использованием бора.
18. Поясните применение алгоритма Ахо – Корасик. Приведите его вычислительную и емкостную сложность.