

Лабораторная работа № 2

по курсу “Языки программирования и методы программирования” (информатика, 3 семестр)

Техническое задание

1. Постановка задачи

Написать программу на C++ для сравнения различных алгоритмов поиска. Сравнение алгоритмов должно производиться на одной из приведенных задач, связанных с обработкой информации. Выполнить реализацию. Написать для нее тесты. Реализовать пользовательский интерфейс.

2. Функциональные требования

Задачи, выполняемые в данной работе: поиск наиболее частой подпоследовательности, размер которой задается пользовательским диапазоном целых чисел, в строке; обработка разреженных векторов; построение алфавитного указателя, где количество слов в строке задается пользователем. Все задания на ассоциативную память.

Задачи должны быть выполнены с помощью реализации шаблонного класса IDictionary, построенного на бинарном дереве, и классе Sequence, написанного в предыдущих лабораторных работах.

Класс SearchMaxSub должен находить наиболее часто повторяющуюся подпоследовательность размера, укладывающуюся в пределы, указанные пользователем, и уметь выводить ее, ее длину, пару строка - длина в качестве ответа.

Класс Sparse должен обрабатывать разреженные векторы, то есть удалять нулевые элементы, сохраняя индексацию для ненулевых элементов, а также выводить как разреженные, так и обработанные вектора.

Класс Split_into_pages разбивает текст на страницы, заданного размера, умеет выводить отдельные страницы, весь обработанный текст, количество страниц, полученных в “новом тексте”.

3. Требования к структурам данных

IDictionary:

Хеш-таблица должна хранить данные в структуре данных, обеспечивающих обращение по индексу. Хеш-функция должна вычислять

положение в этом массиве на основании ключа, а также проверять наличие элемента в таблице по ключу. При удалении элементов в таблице должны отсутствовать пустоты. Таблица обязана поддерживать добавление новых элементов и знать о количестве элементов, хранимых в ней.

BinTree

Класс бинарного дерева должен поддерживать удаление, добавление, поиск и проверку наличия элементов в структуре данных, нахождение поддеревя, поиск минимального и максимального элементов, вывод в консоль, удаление всего дерева и запоминание его размера.

pair_

Класс “пара” должен поддерживать операторы сравнения для различных пар, вывод в консоль, конструктор по ключу и значению, пустой конструктор и по ссылке на другую пару.

Sequence

Класс, являющийся “интерфейсом” классов ArraySequence и LinkedListSequence, должен поддерживать такие функции как получение значения по индексу, вставка в любое место последовательности, изменение размера, получения подпоследовательности, удаление элемента, добавление или замена элемента по индексу.

SortSequence

Класс должен поддерживать те же функции, что и стандартный Sequence, но являться отсортированной последовательностью. Должен присутствовать конструктор, позволяющий выбрать метод сортировки массива, по которому строится SortSequence.

4. Требования к интерфейсу

Интерфейс должен предоставлять пользователю выбор между задачами, поддерживать как автоматический, так и ручной ввод данных, давать возможность пользователю увидеть результат и промежуточные значения.

5. Требования к входным и выходным данным

Данные должны являться символами или целыми числами в зависимости от выбранной задачи.

6. Требования к тестированию

Все основные алгоритмы должны быть покрыты тестами.