

Задача №7

Сбалансированные деревья, хеш–таблицы

Цель работы – освоение работы с хеш-таблицами, сравнение эффективности поиска в сбалансированных (AVL) деревьях, в двоичных деревьях поиска и в хеш-таблицах. Сравнение эффективности устранения коллизий при внешнем и внутреннем хешировании.

Задание: Построить хеш-таблицу и AVL-дерево по указанным данным. Сравнить эффективность поиска в сбалансированном двоичном дереве, в двоичном дереве поиска и в хеш-таблице (используя открытую и закрытую адресацию). Вывести на экран деревья и хеш-таблицы. Подсчитать среднее количество сравнений для поиска данных в указанных структурах. Произвести реструктуризацию хеш-таблицы, если количество сравнений при поиске/добавлении больше указанного. Оценить эффективность использования этих структур (по времени и по памяти) для поставленной задачи. Оценить эффективность поиска в хеш-таблице при различном количестве коллизий и при различных методах их разрешения.

№ варианта – № по списку % 7 + 1

1. Сбалансировать дерево (задача №6) после удаления повторяющихся букв. Вывести его на экран в виде дерева. Составить хеш-таблицу, содержащую буквы и количество их вхождений во введенной строке. Вывести таблицу на экран. Реализовать операции добавления и удаления введенной буквы во всех структурах. Осуществить поиск введенной буквы в двоичном дереве поиска, в сбалансированном дереве и в хеш-таблице. Сравнить время поиска, объем памяти и количество сравнений при использовании различных структур данных.
2. Используя предыдущую программу (задача №6), сбалансировать полученное дерево. Вывести его на экран в виде дерева. Построить хеш-таблицу из чисел файла. Реализовать операции добавления и удаления введенного числа во всех структурах. Осуществить поиск введенного целого числа в двоичном дереве поиска, в сбалансированном дереве и в хеш-таблице. Сравнить время поиска, объем памяти и количество сравнений при использовании различных структур данных.
3. Построить дерево поиска из слов текстового файла (задача №6), сбалансировать полученное дерево. Вывести его на экран в виде дерева. Удалить все слова, начинающиеся на указанную букву, в исходном и сбалансированном дереве. Построить хеш-таблицу из слов текстового файла. Вывести построенную таблицу слов на экран. Осуществить поиск и удаление введенного слова. Выполнить программу для различных размерностей таблицы и сравнить время удаления, объем памяти и количество сравнений при использовании сбалансированных деревьев и хеш-таблиц.

4. Построить дерево поиска из слов текстового файла, сбалансировать полученное дерево. Вывести его на экран в виде дерева. Удалить все слова, начинающиеся на указанную букву, в исходном и сбалансированном дереве. Сравнить время удаления, объем памяти. Построить хеш-таблицу из слов текстового файла. Вывести построенную таблицу слов на экран. Осуществить поиск и удаление введенного слова, вывести таблицу. Выполнить программу для различных размерностей таблицы и сравнить время удаления, объем памяти и количество сравнений при использовании сбалансированных деревьев и хеш-таблиц
5. Построить хеш-таблицу для зарезервированных слов языка C++ (не менее 20 слов), содержащую HELP для каждого слова. Выдать на экран подсказку по введенному слову. Выполнить программу для различных размерностей таблицы и сравнить время поиска и количество сравнений. Для указанных данных создать сбалансированное дерево. Добавить в таблицы и сбалансированное дерево подсказку по вновь введенному слову, используя при необходимости реструктуризацию таблицы. Сравнить эффективность добавления ключа в сбалансированное дерево и таблицу (с учетом ее реструктуризации).
6. Используя предыдущую программу (задача №6), сбалансировать полученное дерево. Вывести его на экран в виде дерева. Построить хеш-таблицу из слов текстового файла, задав размерность таблицы с экрана. Реализовать операции добавления и удаления слова во всех структурах. Осуществить поиск введенного слова в двоичном дереве поиска, в сбалансированном дереве и в хеш-таблице. Сравнить время поиска, объем памяти и количество сравнений при использовании различных структур данных.
7. Построить дерево поиска из чисел, содержащихся в текстовом файле, сбалансировать полученное дерево. Вывести его на экран в виде дерева. Реализовать операции добавления и удаления числа в деревья. Построить хеш-таблицу из чисел текстового файла. Вывести построенную таблицу чисел на экран. Осуществить поиск и добавление введенного числа. Выполнить программу для различных размерностей таблицы и сравнить время добавления (с учетом реструктуризации при необходимости), объем памяти и количество сравнений при использовании сбалансированных деревьев и хеш-таблиц.