МГТУ им. Н.Э. Баумана

**Лабораторная Работа №6**

**Обработка деревьев, хеш-функций**

**Цель работы**: получить навыки применения двоичных деревьев, реализовать основные операции над деревьями: обход деревьев, включение, исключение и поиск узлов; построить и обработать хеш-таблицы, сравнить эффективность поиска в сбалансированных деревьях, в двоичных деревьях поиска и в хеш-таблицах.

Студент: Нгуен Ань Тхы

Группа : ИУ7 И - 36Б

1) Описание условия задачи:

В текстовом файле содержатся целые числа. Построить ДДП из чисел файла. Вывести его на экран в виде дерева. Сбалансировать полученное дерево и вывести его на экран. Построить хеш-таблицу из чисел файла. Использовать метод цепочек для устранения коллизий. Осуществить поиск введенного целого числа в ДДП, в сбалансированном дереве, в хеш-таблице и в файле. Сравнить время поиска, объем памяти и количество сравнений при использовании различных (4-х) структур данных. Если количество сравнений в хеш-таблице больше указанного (вводить), то произвести реструктуризацию таблицы, выбрав другую функцию.

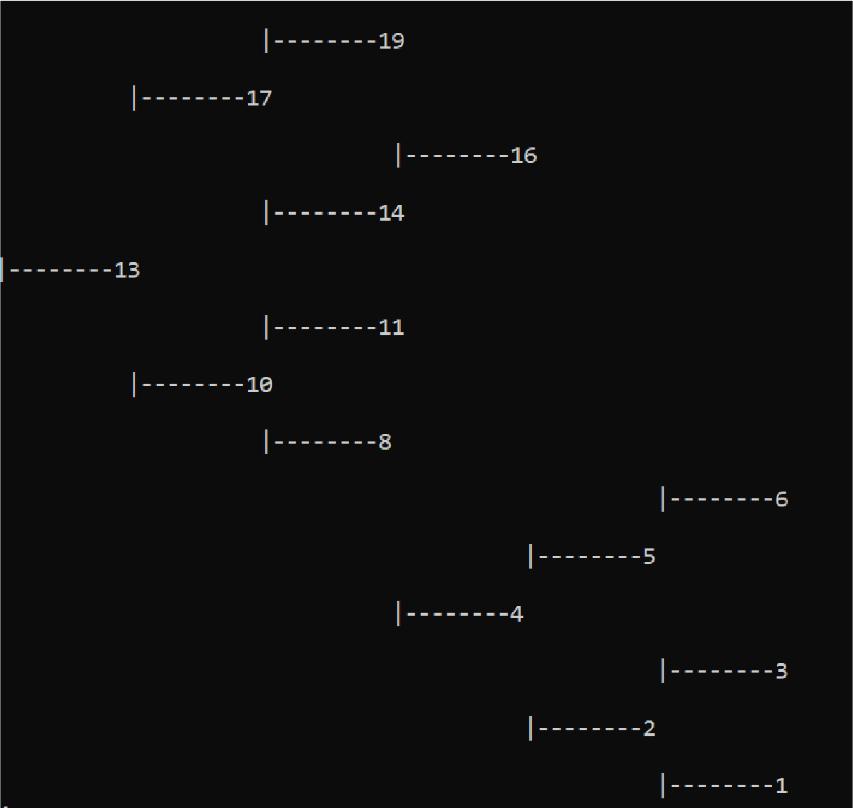
2) Описание ТЗ:

Исходные данные:

Текстовый файл содержаться целые числа.

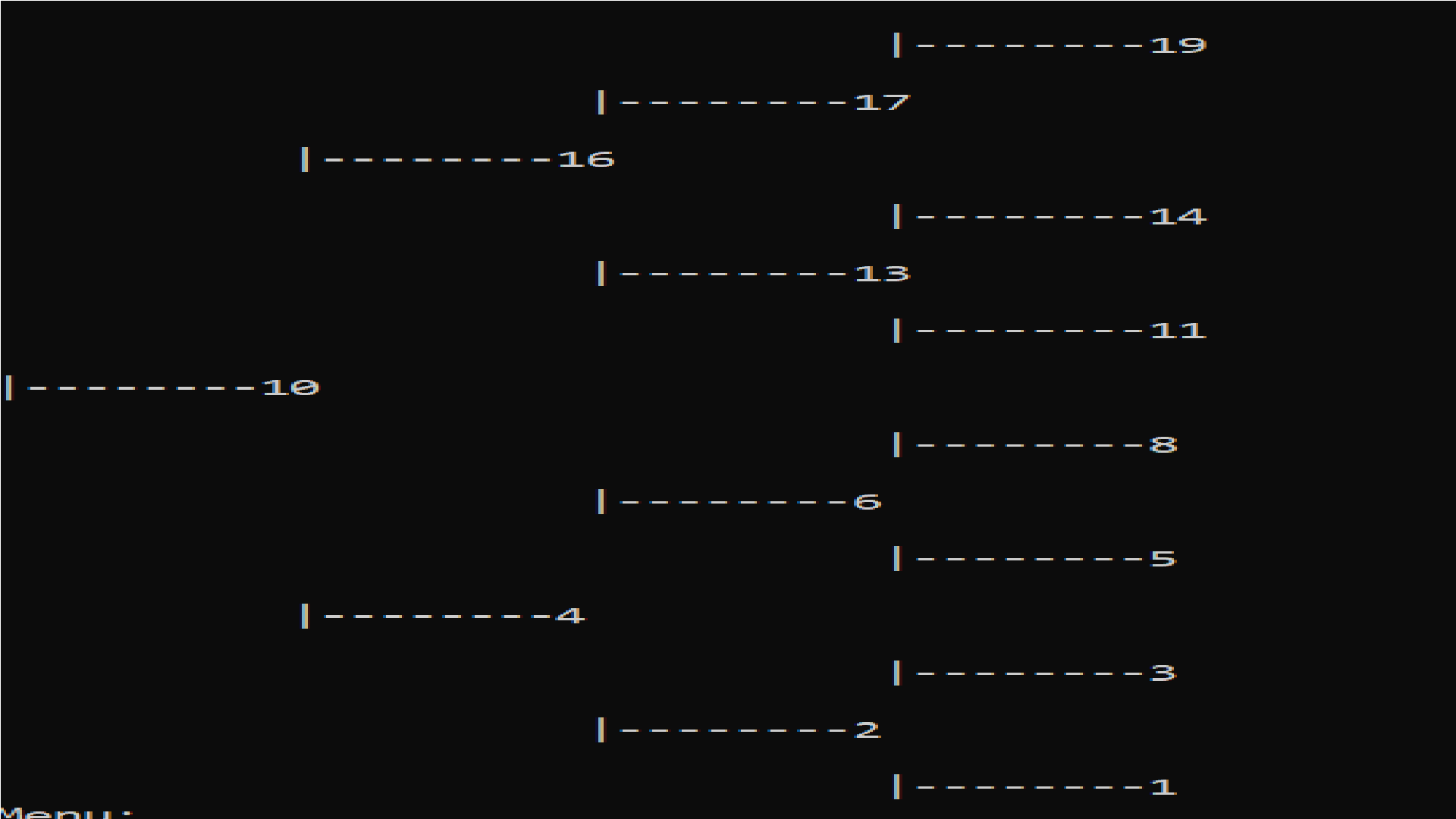
Выходные данные:

Информации соответствует каждому операциями:

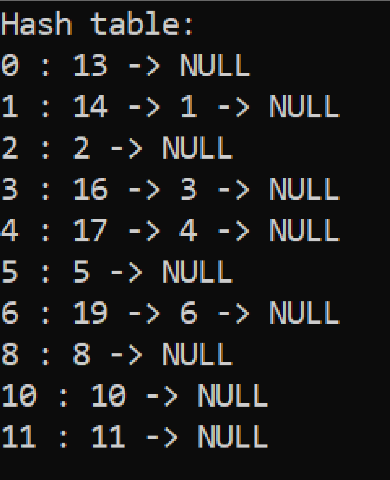
1. ДДП из чисел файла в виде дерева
2. Сбалансированное дерево
3. Хеш-таблицы
4. Результат искать элемент в дереве
5. Сравнение время поиска, объем памяти

3) Пример Тесты:

Начальное дерево:

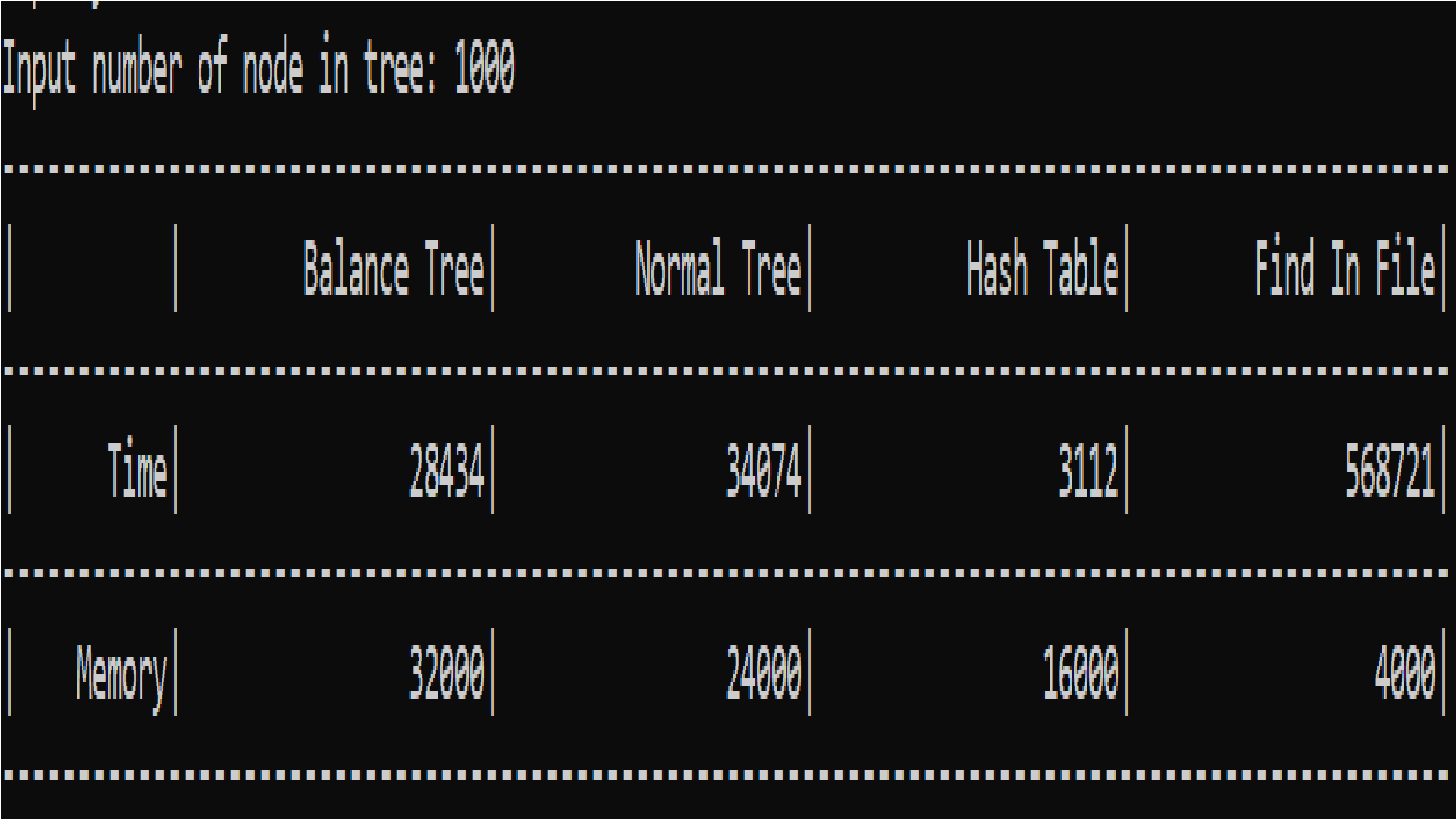


Сбалансированное дерево:



Хеш-таблица:

4) Времени и объем памяти выполнения программы



Вывод:

Самой оптимальной структурой по времени для поиска данных является хеш-таблица. Подобранные хорошо изначально параметры для хеш-таблицы позволяет уменьшить временные затраты на ее перегенерацию и в дальнейшем уменьшить среднее количество сравнений при поиске. Поиск в сбалансированном дереве быстрее, чем в обычном.

5) Контрольные вопросы:

1.Что такое дерево?

Дерево – нелинейная структура данных, которая используется для представления иерархических связей «один ко многим». Дерево с базовым типом Т определяется рекурсивно: это либо пустая структура (пустое дерево), либо узел типа Т с конечным числом древовидных структур того же типа – поддеревьев.

2. Как выделяется память под представление деревьев?

Выделение памяти под деревья определяется типом их представления. Это может быть таблица связей с предками (№ вершины - № родителя), или связный список сыновей. Оба представления можно реализовать как с помощью матрицы, так и с помощью списков. При динамическом представлении деревьев (когда элементы можно удалять и добавлять) целесообразнее использовать списки – т.е. выделять память под каждый элемент динамически.

3. Какие стандартные операции возможны над деревьями?

Обход, поиск, добавление и удаление элемента.

4. Что такое дерево двоичного поиска?

Дерево двоичного поиска – дерево, в котором все левые потомки «моложе» предка, а все правые – «старше». Это свойство выполняется для любого узла, включая корень.

5. Чем отличается идеально сбалансированное дерево от АВЛ дерева?

АВЛ-дерево — сбалансированное по высоте двоичное дерево поиска: для каждой его вершины высота её двух поддеревьев различается не более чем на 1.

Идеально сбалансированное: для каждой его вершины количество вершин в левом и правом поддереве различаются не более чем на 1.

6. Чем отличается поиск в АВЛ-дереве от поиска в дереве двоичного поиска?

Алгоритм поиска одинаковый.

7. Что такое хеш-таблица, каков принцип ее построения?

Массив, заполненный в порядке, определенным хеш-функцией, называется хеш-таблицей.

8. Что такое коллизии? Каковы методы их устранения.

Может возникнуть ситуация, когда разным ключам соответствует одно значение хеш-функции, то есть, когда h(K1) = h(K2), в то время как K1 ≠ K2. Такая ситуация называется коллизией. Для устранения коллизий нужно выбрать другую хеш функцию (или эту же с другими параметрами) и перегенерировать таблицу.

9. В каком случае поиск в хеш-таблицах становится неэффективен?

При большом количестве коллизий.