

### **3. АЛГОРИТМЫ ОБРАБОТКИ ДЕРЕВЬЕВ**

#### **3.1. Задания по построению двоичного дерева и обходам дерева**

1. Разместить в памяти компьютера данное двоичное дерево (см. ниже, номер задания соответствует номеру в журнале группы), данные в вершинах заполнить целыми числами (в диапазоне от 1 до 20).
2. Запрограммировать обходы двоичного дерева сверху вниз, слева направо и снизу вверх, и вывести на экран получившиеся последовательности данных.
3. Для построенного дерева вычислить размер, контрольную сумму, высоту и среднюю высоту. (Для средней высоты предусмотреть вывод двух знаков после запятой).
4. Освоить написание вручную обходов любого заданного двоичного дерева.

#### **3.2. Задания по построению идеально сбалансированного дерева поиска ИСДП**

1. Для набора из 12 первых последовательных неповторяющихся символов ФИО студента построить вручную идеально сбалансированное дерево поиска (ИСДП). При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.
2. Разработать функцию построения ИСДП, построить ИСДП из 100 вершин, распечатать обход дерева слева направо.
3. Для построенного дерева вычислить размер, контрольную сумму, высоту и среднюю высоту. (Для средней высоты предусмотреть вывод двух знаков после запятой).
- 4\*. Реализовать графическую функцию изображения ИСДП на экране.

#### **2.3. Задания по построению случайного дерева поиска СДП**

1. Для набора из 12 первых последовательных неповторяющихся символов ФИО студента построить вручную случайное дерево поиска (СДП). При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.
2. Разработать две функции добавления вершины в случайное дерево поиска: рекурсивно и с двойной косвенностью.
3. Для одной последовательности из 100 вершин, заданных случайно, построить два случайных дерева поиска: рекурсивно и с двойной косвенностью. Распечатать обходы слева направо для построенных деревьев.

4. Для построенных деревьев вычислить размер, контрольную сумму, высоту и среднюю высоту, сравнить их с аналогичными характеристиками ИСП, построить таблицу вида:

n=100	Размер	Контр. сумма	Высота	Средн.высота
ИСП				
СП1				
СП2				

5\*. Реализовать графическую функцию изображения случайного дерева поиска на экране.

#### **2.4. Задания по удалению из случайного дерева поиска СП**

1. Для построенного из 12 первых последовательных неповторяющихся символов ФИО студента случайного дерева поиска (СП) выполнить удаление всех вершин (в том же порядке, как и при построении). При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.

2. Разработать функцию удаления из случайного дерева поиска (СП) вершины с заданным ключом.

3. Реализовать удаление из СП 10 вершин, задаваемых с клавиатуры, распечатывать обход дерева слева направо (или изображение дерева) после каждой удаленной вершины.

4\*. На базе случайного дерева поиска построить словарь частот встречаемости ключевых слов в тексте программы на Си, вывести на экран ключевые слова и их частоты.

#### **2.5. Задания по построению АВЛ-дерева поиска**

1. Для набора из 12 первых последовательных неповторяющихся символов ФИО построить вручную АВЛ-дерево поиска. При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.

2. Разработать функцию добавления вершины в АВЛ-дерево поиска.

3. Для последовательности из 100 вершин, заданных случайно, построить АВЛ-дерево поиска. Распечатать обход слева направо для построенного дерева.

4. Для построенного AVL-дерева вычислить размер, контрольную сумму, высоту и среднюю высоту, сравнить их с аналогичными характеристиками ИСП, построить таблицу вида:

n=100	Размер	Контр. Сумма	Высота	Средн.высота
ИСП				
AVL				

5\*. Реализовать графическую функцию изображения AVL-дерева на экране.

## 2.6. Задания по удалению из AVL-дерева поиска

1. Для построенного из 12 первых последовательных неповторяющихся символов ФИО студента AVL-дерева поиска выполнить удаление всех вершин (в том же порядке, как и при построении). При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.

2. Разработать функцию удаления из AVL-дерева поиска вершины с заданным ключом.

3. Реализовать удаление из AVL-дерева 10 вершин, задаваемых с клавиатуры, распечатывать обход дерева слева направо (или изображение дерева) после каждой удаленной вершины.

4\*. Подтвердить экспериментально утверждение, что при добавлении вершин в AVL-дерево поиска на каждые два включения встречается один поворот, а при удалении вершин поворот происходит в одном случае из пяти.

## 2.7. Задания по построению двоичного Б-дерева поиска

1. Для набора из 12 первых последовательных неповторяющихся символов ФИО построить вручную двоичное Б-дерево поиска (ДБД). При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.

2. Разработать функцию добавления вершины в двоичное Б-дерево поиска (ДБД).

3. Для последовательности из 100 вершин, заданных случайно, построить двоичное Б-дерево поиска. Распечатать обход слева направо для построенного дерева.

4. Для построенного ДБД вычислить размер, контрольную сумму, высоту и среднюю высоту, сравнить их с аналогичными характеристиками AVL-дерева, построить таблицу вида:

n=100	Размер	Контр. Сумма	Высота	Средн.высота
АВЛ				
ДБД				

5\*. Разработать функцию вычисления количества уровней двоичного Б-дерева.

6\*. Реализовать графическую функцию изображения двоичного Б-дерева на экране с учетом горизонтального расположения страниц дерева.

## 2.8. Задания по построению дерева оптимального поиска (точный алгоритм)

1. Для набора из 4 первых последовательных неповторяющихся символов ФИО построить вручную дерево оптимального поиска (ДОП) с помощью точного алгоритма (вес символа – количество встреч этого символа в ФИО), Вычислить средневзвешенную высоту построенного дерева. При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.

2. Разработать функцию построения дерева оптимального поиска (точный алгоритм). Предусмотреть вывод на экран матрицы весов, матрицы взвешенных высот и матрицы корней поддеревьев.

3. Для последовательности из 100 упорядоченных вершин построить дерево оптимального поиска (ДОП) (вес вершины задать случайно в диапазоне от 1 до 100). Распечатать обход слева направо для построенного дерева. Для проверки правильности точного алгоритма сравнить  $AP[0,n]/AW[0,n]$ , взятое из матриц, со средневзвешенной высотой построенного дерева.

4. Для построенного ДОП вычислить размер, контрольную сумму, высоту и средневзвешенную высоту, построить таблицу вида:

n=100	Размер	Контр. Сумма	Высота	Средневзвеш.высота
ДОП				

5\*. Реализовать графическую функцию изображения дерева оптимального поиска (ДОП) на экране с указанием веса каждой вершины.

## 2.9. Задания по построению дерева оптимального поиска (приближенные алгоритмы)

1. Для набора из 12 первых последовательных неповторяющихся символов ФИО построить вручную почти оптимальные деревья поиска с помощью

приближенных алгоритмов A1 и A2 (вес символа – количество встреч этого символа в ФИО), Вычислить и сравнить средневзвешенные высоты построенных деревьев. При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.

2. Реализовать приближенные алгоритмы A1 и A2 для построения почти оптимальных деревьев поиска..

3. Для последовательности из 100 вершин построить почти оптимальные деревья поиска с помощью приближенных алгоритмов A1 и A2 (веса вершин в диапазоне от 1 до 100 совпадают с аналогичными для построенного ДОП).. Распечатать обходы слева направо для построенных деревьев.

4. Для построенных почти оптимальных деревьев вычислить размер, контрольную сумму, высоту и средневзвешенную высоту, сравнить их с аналогичными характеристиками дерева оптимального поиска, построить таблицу вида:

n=100	Размер	Контр. Сумма	Высота	Средневзвеш.высота
ДОП				
A1				
A2				

5\*. Реализовать графическую функцию изображения почти оптимального дерева поиска на экране с указанием веса каждой вершины.