**Содержание**

[Тест №1 2](#_Toc104377204)

[Тест №2 3](#_Toc104377205)

[Тест №3 5](#_Toc104377206)

[Кроссворд 8](#_Toc104377207)

[Вопросы к кроссворду 9](#_Toc104377208)

[Раздел 1 10](#_Toc104377209)

[Тема 1.1 Введение в дисциплину. Общие понятия и определения. 10](#_Toc104377210)

[Тема 1.2 Статические структуры данных. 11](#_Toc104377211)

[Раздел 2 13](#_Toc104377212)

[Тема 2.1 Список. 13](#_Toc104377213)

[Тема 2.2 Стек. 14](#_Toc104377214)

[Тема 2.3 Очередь 15](#_Toc104377215)

[Раздел 3 17](#_Toc104377216)

[Тема 3.1 Деревья. 17](#_Toc104377217)

[Тема 3.2 АТД «Дерево». 19](#_Toc104377218)

[Тема 3.3 Основные виды деревьев 22](#_Toc104377219)

[Список используемой литературы 24](#_Toc104377220)

# Тест №1

1. Что такое класс?
2. Это тип данных, задающий реализацию некоторой абстракции данных, характерной для проблемной области, в интересах которой создается программная система.
3. Тип данных, сформированный из объектов однородных либо разнообразных типов данных.
4. Это массив, каждый элемент которого сам является одномерным массивом.
5. Во внутреннем представлении, это целочисленный тип данных, только здесь пользователь вместо числа использует заранее определенные строковые значения.
6. Укажите какой вариант не относится к простым структурам данных.
7. Числовой
8. Логический
9. Кольцевой
10. Символьный
11. Что такое ссылка?
12. Переменная, значением которой является адрес ячейки памяти.
13. Это объект, указывающий на определенные данные, но не хранящий их.
14. Изменение числа элементов и (или) связей между элементами структуры.
15. Один из видов наказаний в Российской Империи.

4. Какая операция не может быть выполнена над структурами данных?

1. Создание
2. Обновление
3. Уничтожение
4. Сортировка

5. Операция обновления…

1. …позволяет изменить значения данных в структуре данных.
2. …используется программистами для доступа к данным внутри самой структуры.
3. …заключается в выделении памяти для структуры данных.
4. …противоположна по своему действию операции создания.

6. Что из перечисленного не является операцией совершаемой над записями?

1. Операция квалификации
2. Операция выборки
3. Операция присваивания
4. Операция сравнения

7. Что такое запись?

1. Совокупность ограниченного числа логически связанных компонент, принадлежащих к разным типам.
2. Название некоторых официальных документов. Дарственная запись. Духовная запись.
3. То, что является письменной передачей, изложением чего-либо.
4. Занесенные на бумагу мысли, замечания.

8. Что не содержит описание массива?

1. Имя
2. Тип элементов
3. Диапазоны изменения индексов
4. Таблица

9. Что не относится к структурированным типам?

1. Массив
2. Запись
3. Множество
4. Очередь

10. Какая из операций не может быть выполнена над множеством?

1. Создание
2. Найти элемент
3. Сравнение
4. Выборка

# Тест №2

1. Что такое динамическая структура данных?

1. Это структура данных, память под которую выделяется и освобождается по мере необходимости.
2. Это линейная динамическая структура данных, у каждого элемента может быть только один предок и только один потомок.
3. Это структура данных, представляющая собой последовательность элементов, в каждом из которых хранится значение и указатель на следующий элемент.
4. Элемент структуры, идущий после текущего.

2. Как называются элементы структуры идущие после первого?

1. Конец
2. Последний вагон
3. Хвост
4. Голова

3. Чем отличается список от массива?

1. Количеством элементов
2. Размер списка не имеет ограничений
3. Возможными операциями
4. Названием

4. Какие виды списков бывают?

1. Кольцевой
2. Длинный
3. Односвязный
4. Двусвязный

5. Что не является операцией совершаемой над списком?

1. Выборка
2. Вставка элемента
3. Удаление элемента
4. Проверка пустоты

6. Какой принцип обслуживания использует очередь?

1. LIFO
2. FIFA
3. FIFO
4. LIL PEEP

7. Какого вида очереди не существует?

1. Простая
2. Кольцевая
3. Очередь с приоритетами
4. Двойная

8. С помощью чего нельзя реализовать очередь?

1. Кольцевого списка
2. Односвязного списка
3. Двусвязного списка
4. Динамического массива

9. Что такое дек?

1. Это динамическая структура данных, добавление элементов в которую выполняется в один конец, а выборка — из другого конца.
2. Это структура данных, представляющая собой последовательность элементов, в которой можно добавлять и удалять в произвольном порядке элементы с двух сторон.
3. Это структура данных, представляющая собой последовательность элементов, в каждом из которых хранится значение и указатель на следующий элемент.
4. Это линейная динамическая структура данных, у каждого элемента может быть только один предок и только один потомок.

10. Какая из операций не может быть совершена над деком?

1. Очистка
2. Создание
3. Добавление элемента в середину
4. Проверка пустоты

# Тест №3

1. Что не относится к линейным структурам данных?

1. Леса
2. Деревья
3. Массивы
4. Графы

2. Что такое дерево?

1. Это объект, указывающий на определенные данные, но не хранящий их.
2. Это тип данных, задающий реализацию некоторой абстракции данных, характерной для проблемной области, в интересах которой создается программная система.
3. Это совокупность элементов, называемых узлами, и отношений ("родительских"), образующих иерархическую структуру узлов.
4. Совокупность ограниченного числа логически связанных компонент, принадлежащих к разным типам.

3. Каким символом обозначается нулевое дерево?

1. Т0
2. Д0
3. Ω
4. Λ

4. Что такое высота узла дерева?

1. Длина самого длинного пути из этого узла до какого-либо листа.
2. Определяется как длина пути от корня до этого узла.
3. Число, на единицу меньшее числа узлов, составляющих этот путь.
4. Последовательность узлов

5. Какого вида обхода деревьев не существует?

1. Прямой
2. Кольцевой
3. Обратный
4. Симметричный

6. Что такое АВЛ-дерево?

1. Сбалансированное двоичное дерево поиска с k=1.
2. Структура данных, являющаяся B-деревом Степени 1, страницы которого могут содержать только 2-вершины (вершины с одним полем и 2 детьми) и 3-вершины (вершины с 2 полями и 3 детьми).
3. Это особый тип сбалансированного дерева поиска, в котором каждый узел может содержать более одного ключа и иметь более двух дочерних элементов.
4. Структура данных на основе B-дерева, сбалансированное n-арное дерево поиска с переменным, но зачастую большим количеством потомков в узле.

7. Что не соответствует свойствам красно-чёрного дерева?

1. Каждый узел окрашен либо в красный, либо в черный цвет (в структуре данных узла появляется дополнительное поле – бит цвета).
2. Корень окрашен в красный цвет.
3. Листья(так называемые NULL-узлы) окрашены в черный цвет.
4. Пути от узла к его листьям должны содержать одинаковое количество черных узлов(это черная высота).

8. Что не является свойством 2-3 дерева?

1. Все нелистовые вершины содержат одно поле и 2 поддерева или 2 поля и 3 поддерева.
2. Все листовые вершины находятся на одном уровне (на нижнем уровне) и содержат 1 или 2 поля.
3. Все данные отсортированы (по принципу двоичного дерева поиска).
4. Нелистовые вершины содержат более двух полей, указывающие на диапазон значений в их поддеревьях.

9. Какие типы обходов деревьев не существуют?

1. В высоту
2. В ширину
3. Прямой в глубину
4. Обратный в глубину

10. Что такое дерево Хаффмана?

1. Тип дерева с наименьшей длиной пути веса.
2. Древовидная структура данных, в которой у родительских узлов не может быть больше двух детей.
3. Особый тип бинарного дерева, в котором у каждого внутреннего узла по два ребенка, а листовые вершины находятся на одном уровне.
4. Дерево, в котором на каждый уровень приходится по одной вершине.

# Кроссворд

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | **7** |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | **9** |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **2** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **4** |  |  |  |  |
|  |  |  | **5** |  |  |  | **3** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **8** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **6** |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | **10** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | **7** |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | **9** | з | а | ч | е | т |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | л |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | г |  |  |  |  |  |  |  |  |  |  |  |
|  | **2** |  |  |  |  |  |  |  | о |  |  |  |  |  |  | **4** |  |  |  |  |
|  | с |  | **5** |  |  |  | **3** |  | р |  |  |  |  |  |  | д |  |  |  |  |
| **1** | п | р | о | г | р | а | м | м | и | р | о | в | а | н | и | е |  |  |  |  |
|  | и |  | ч |  |  |  | а |  | т |  |  |  |  |  |  | р |  |  |  |  |
| **8** | с | т | е | к |  |  | т |  | м |  |  |  |  |  |  | е |  |  |  |  |
|  | о |  | р |  |  |  | р |  |  |  |  |  |  |  |  | в |  |  |  |  |
|  | к |  | е |  |  |  | и |  |  |  |  |  |  |  | **6** | о | б | х | о | д |
|  |  |  | д |  |  |  | ц |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | ь |  | **10** | м | а | с | с | и | в |  |  |  |  |  |  |  |  |  |

# Вопросы к кроссворду

1. Процесс создания компьютерных программ.
2. Это линейная динамическая структура данных, у каждого элемента может быть только один предок и только один потомок.
3. Двумерный массив.
4. Одна из наиболее широко распространённых структур данных в информатике, эмулирующая древовидную структуру в виде набора связанных узлов.
5. Это структура данных, в которой реализуется принцип First-in, First-out.
6. Бывает прямой, обратный и симметричный.
7. Система последовательных операций (в соответствии с определёнными правилами) для решения какой-н. задачи.
8. Абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO.
9. Что я хочу получить по АиСД?
10. Структура данных, хранящая набор значений, идентифицируемых по индексу или набору индексов, принимающих целые значения из некоторого заданного непрерывного диапазона.

# Раздел 1

# Тема 1.1 Введение в дисциплину. Общие понятия и определения.

Алгоритмы – это система последовательных операций (в соответствии с определёнными правилами) для решения какой-либо. задачи. Понятие «алгоритм» является основополагающим для программ. Углубимся в анализ этого термина. Слово «алгоритм» (algorithm) уже само по себе представляет большой интерес. Этот термин исходит от слова «algorism», которое означает «выполнение арифметических действий с помощью арабских цифр». Слово «алгоритм» у некоторых может ассоциироваться с алгоритмом Евклида, который представляет собой процесс нахождения общего делителя двух чисел. Данный алгоритм был описан еще в 300 году до нашей эры. Современное значение слова «алгоритм» во многом аналогично таким понятиям как процесс, рецепт, метод, процедура, способ, программа.

Структура данных – это множество элементов данных и внутренних связей между ними. Структуры данных делятся на простые (базовые, примитивные) и интегрированные (сложные, структурированные, композитные) структуры данных.

К простым структурам данных относятся: числовые, битовые, символьные, логические, указатели. Простыми называются структуры данных, которые не могут быть разделены на составные части, большие, чем биты.

Интегрированные структуры данных организуются из простых и других интегрированных структур.

Также следует различать физические и логические структуры данных.

Физическая структура данных – это ее представление в памяти в том виде, как она выглядит «на самом деле».

Логическая структура данных – это созданная программными средствами образное, абстрактное ее представление.

По наличию связей структуры классифицируют на: несвязные (массивы, векторы, строки, стеки, очереди), связные (связные списки).

Также существует понятие изменчивости — это изменение количества элементов либо связей между ними. По признаку изменчивости структуры бывают: статические, полустатические, динамические

Отдельного упоминания заслуживают файлы как структуры данных. Файлы бывают: последовательные, прямого или комбинированного доступа, организованные разделами.

Следующий не менее важный критерий структур данных – это характеристика упорядоченности элементов. По упорядоченности структуры бывают: нелинейные (деревья, графы, многосвязные списки), линейные (строки, векторы, массивы, стеки, очереди, односвязные и двусвязные списки).

Над любыми структурами данных могут выполняться четыре общие операции: создание, удаление, доступ, обновление. Эти операции являются обязательными для всех структур данных. Но также для каждой структуры данных могут быть определены операции специфические, работающие только с данной структурой. Эти специфические операции рассматриваются при рассмотрении конкретных структур данных.

Знание структуры данных позволяет организовать их хранение и обработку максимально эффективным образом - с точки зрения минимизации затрат как памяти, так и процессорного времени. Другим не менее важным преимуществом, которое обеспечивается структурным подходом к данным, является возможность структурирования сложного программного изделия. При структурировании больших программных изделий возможно применение подхода, основанного на структуризации алгоритмов и известного, как "нисходящее" проектирование, "программирование сверху вниз", или подхода, основанного на структуризации данных и известного, как "восходящее" проектирование, "программирование снизу вверх".

# Тема 1.2 Статические структуры данных.

Статические структуры данных – это структурированное множество простых структур. Для статических структур несвойственна изменчивость. Размер памяти, которые отводится для данных этой структуры является константным и выделяется на этапе компиляции (или выполнения) программы.

**Вектор**

Вектором называется обыкновенный одномерный массив. Это структура данных, где число элементов фиксировано, причем речь идет об однотипных компонентах. У каждого элемента вектора есть свой уникальный индекс. С физической точки зрения векторные компоненты размещаются в памяти в ячейках, расположенных последовательно (подряд).

**Массивы**

Массив – это упорядоченная последовательность данных, которая состоит из конечного числа элементов, имеющих один и тот же тип. Все элементы обозначаются одним именем и имеют свой уникальный индекс.

Двумерный массив (матрица) представляет собой вектор, каждый элемент которого тоже является вектором. Все, что является справедливым для вектора, является справедливым и для матрицы.

Свободным называется двухмерный массив (матрица), размер строк которого может быть различным. Преимущество использования свободного массива заключается в том, что не требуется отводить память компьютера с запасом для размещения строки максимально возможной длины.

**Записи**

Запись – это комбинированный тип данных, в котором значения представляют собой нетривиальную структуру. Записи формируются из нескольких полей разного типа, внешний доступ к этим полям происходит по именам полей. Записи – это средство представления программных моделей реальных объектов.

В памяти компьютера записи можно представить как последовательность полей, которые занимают произвольную непрерывную область памяти или в виде связного списка, имеющего указатели на значения полей записи.

**Множества**

Множество – это набор неповторяющихся данных одного типа. Множество способно принимать все значения базового типа, а так как он не должен превышать 256 значений, то типом элементов множеств могут быть char, byte и их производные. В памяти множество хранится в виде массива битов, причем каждый бит показывает, принадлежит ли элемент объявленному множеству. Максимальное количество элементов множества – 256. Само множество не может превышать 32 байта.

# Раздел 2

Динамические структуры данных – это структуры данных, память под которые выделяется и освобождается по мере необходимости.

Динамическое распределение памяти - память под отдельные элементы выделяется в момент, когда они "начинают существовать" в процессе выполнения программы, а не во время компиляции.

Динамические структуры данных можно охарактеризовать тем, что: они не имеют имени, память выделяется в процессе выполнения программы, количество элементов структуры может изменяться, размерность структуры может изменяться, характер взаимосвязи между элементами структуры может изменяться.

К динамическим структурам данных относятся: однонаправленные, двунаправленные, кольцевые однонаправленные и кольцевые двунаправленные, стеки, деки, очереди, деревья и другие.

# Тема 2.1 Список.

Список — это линейная динамическая структура данных, у каждого элемента может быть только один предок и только один потомок. Список похож на обыкновенный массив. Но размер списка неограничен.

Экземпляр списка является компьютерной реализацией математического понятия конечной последовательности (пронумерованный набор каких-либо объектов, среди которых допускаются повторения, причём порядок объектов имеет значение).

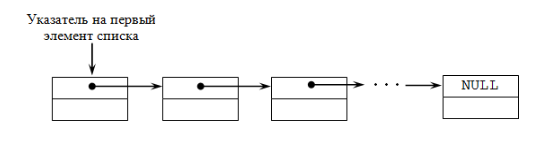
Операторы АТД «Список»: вставка элемента, поиск элемента, вывод значения элемента, удаление элемента, уничтожение списка, печать первого (последнего) и всех элементов списка.

Рисунок 1

Рассмотрим реализацию списка. На рисунке 1 показан односвязный список. Каждый элемент списка имеет указатель на следующий элемент. Последний элемент списка указатель на следующий элемент отсутствует (NULL). Все элементы идут последовательно друг за другом.

В языке программирования C++ массивы, по сути своей и являются списками.

Списки в зависимости от связей бывают: односвязные, двусвязные, циклические.

Однонаправленный (односвязный) список — это структура данных, состоящая из элементов одного типа, связанных между собой последовательно посредством указателей. Каждый элемент списка имеет указатель на следующий элемент.

Двусвязный список очень похож на односвязный. Но каждый узел двунаправленного (двусвязного) линейного списка (ДЛС) содержит два поля указателей — на следующий и на предыдущий узлы. Указатель на предыдущий узел корня списка содержит нулевое значение. Указатель на следующий узел последнего узла также содержит нулевое значение.

Циклический (кольцевой) список – это структура данных, представляющая собой последовательность элементов, последний элемент которой содержит указатель на первый элемент списка, а первый (в случае двунаправленного списка) – на последний.

Преимущества списков проявляются в тех структурах данных, где операции изменения порядка превалируют над операциями доступа и поиска. В качестве примеров алгоритмов, в которых используется список можно назвать алгоритмы изменения порядка или сортировки.

# Тема 2.2 Стек.

Стек - абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (Last in – First out). Стек состоит из ячеек, которые представлены в виде структуры, содержащей какие-либо данные и указатель типа данной структуры на следующий элемент. Таким образом стек является разновидностью списка.

Стек – это список, у которого доступен один элемент. Этот элемент называется вершиной стека. Взять и добавить элемент можно только в вершину стека. Если в стек добавлены числа 1,2,3 то при последующем извлечении этих элементов мы получим 3,2,1.

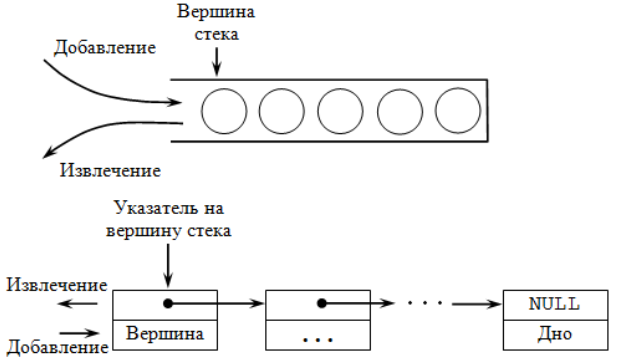
Основные операции, производимые над стеком: создание стека, печать (просмотр) стека, добавление элемента в вершину стека, извлечение элемента из вершины стека, проверка пустоты стека, очистка стека.

Рисунок 2

Стековый калькулятор - это калькулятор, который накапливает последовательность операндов и операций в стеке, а затем поочерёдно извлекает их из стека и вычисляет результат выражения.

Такой калькулятор удобно использовать, когда надо выполнить операцию не с двумя числами, а вычислить выражение, которое может содержать несколько знаков операций (+, -, \*, / и т.п.), а также скобки, которые определяют очерёдность вычисления математических операций.

Для создания калькулятора, умеющего вычислять выражения, лучше использовать стек и специальный алгоритм помещения данных в стек и извлечения данных из стека, который позволит без особых умственных усилий и сложных способов разбора выражения-строки получить нужный результат, то есть вычислить выражение независимо от того, сколько в этом выражении будет скобок, знаков операций, и в какой последовательности пользователь всё это будет вводить.

# Тема 2.3 Очередь

Очередь — это динамическая структура данных, добавление элементов в которую выполняется в один конец, а выборка — из другого конца. Очередь реализует принцип обслуживания FIFO (First in – First out). В очереди доступны только начало и конец очереди. Поместить элемент можно только в конец очереди, а взять только из ее начала. Примером может служить обыкновенная очередь в магазине.

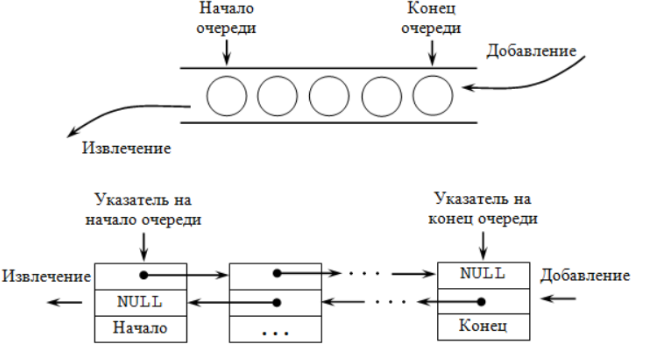
Очереди бывают: простые, кольцевые, с приоритетами.

Рисунок 3

Очередь с приоритетами может быть реализована следующими способами:

1. Очередь с приоритетным включением (элемент, который добавляется в очередь, сразу размещается в ней в соответствии с его приоритетом).
2. Очередь с приоритетным исключением (новый элемент просто добавляется в конец очереди. Вытягивание элемента из очереди осуществляется по его приоритету, то есть вытягивается элемент с наивысшим приоритетом).

Операции над очередями: создание очереди, добавление элемента в очередь (в хвост), удаление элемента из очереди (из головы), проверка на пустоту, очистка очереди.

Дек является особым видом очереди. Дек - это структура данных, представляющая собой последовательность элементов, в которой можно добавлять и удалять в произвольном порядке элементы с двух сторон.

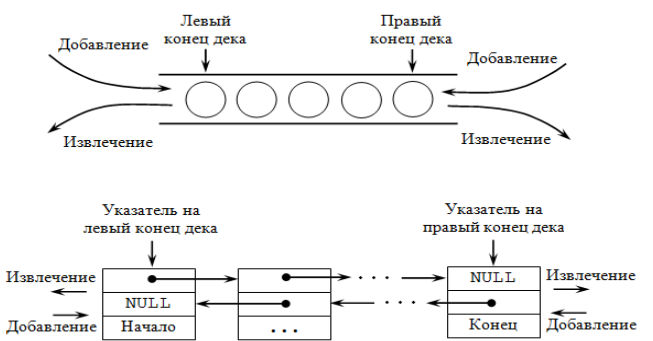
Основные операции с деком: создание дека, печать дека, добавление элемента в левый (правый) конец дека, извлечение элемента из левого (правого) конца дека, проверка пустоты дека, очистка дека.

Рисунок 4

# Раздел 3

# Тема 3.1 Деревья.

Нелинейные структуры данных (деревья, графы, леса) позволяют выражать более сложные отношения между элементами, нежели линейные отношения соседства, как это было в линейных списках.

Дерево — одна из наиболее широко распространённых структур данных в информатике, эмулирующая древовидную структуру в виде набора связанных узлов. Является связным графом, не содержащим циклы. Деревья представляют собой иерархическую структуру некой совокупности элементов.

Примерами деревьев могут служить организационные диаграммы. Деревья используются при анализе электрических цепей, при представлении структур математических формул. деревья используются для организации информации в системах управления базами данных и для представления синтаксических структур в компиляторах программ.

Дерево - это совокупность элементов, называемых узлами (один из которых определен как корень), и отношений ("родительских"), образующих иерархическую структуру узлов.

К основным терминам, связанным с деревьями можно отнести высоту узлов деревьев, высоту самого дерева, глубину узлов.

Высотой узла дерева называется длина самого длинного пути из этого узла до какого-либо листа.

Высота дерева совпадает с высотой корня.

Глубина узла определяется как длина пути (он единственный) от корня до этого узла.

Потомки узла упорядочиваются слева направо (упорядоченное дерево). Если порядок потомков не учитывается, то такое дерево называется неупорядоченным.

Есть несколько способов обхода деревьев: в прямом порядке, в обратном порядке и во внутреннем порядке (симметричный обход).

При прохождении в прямом порядке (т.е. при прямом упорядочивании) узлов дерева Т сначала посещается корень n, затем узлы поддерева Т1, далее все узлы поддерева Т2 и т.д. Последними посещаются узлы поддерева Тk.

При симметричном обходе узлов дерева Т сначала посещаются в симметричном порядке все узлы поддерева Т1, далее корень n, затем последовательно в симметричном порядке все узлы поддеревьев Т2, …, Тk.

Во время обхода в обратном порядке сначала посещаются в обратном порядке все узлы поддерева Т1, затем последовательно посещаются все узлы поддеревьев Т2, …, Тk, также в обратном порядке, последним посещается корень n.

Помеченное дерево – это дерево, вершинам которого присваиваются уникальные номера от 1 до n. Мы можем посчитать такие деревья для малых значений n вручную, чтобы предположить общую формулу. Число помеченных деревьев из n вершин равно nn−2. Два помеченных дерева изоморфны, если их графы изоморфны и соответствующие точки двух деревьев имеют одинаковые метки.

Немеченое дерево – это дерево, вершинам которого не назначены никакие числа.

Деревья выражений представляют код в виде древовидной структуры, где каждый узел является выражением, например, вызовом метода или двоичной операцией, такой как x < y.

Часто при обходе деревьев составляется список не имен узлов, а их меток.

В случае дерева выражений при прямом упорядочивании получаем известную префиксную форму выражений, где оператор предшествует и левому, и правому операндам.

Обратное упорядочивание меток дерева выражений дает так называемое постфиксное представление выражений.

При симметричном обходе дерева выражений мы получим так называемую инфиксную форму выражения, которая совпадает с привычной нам формой записи выражений, но не использует скобок.

# Тема 3.2 АТД «Дерево».

Дерево — это, по сути, один из частных случаев графа. Древовидная модель может быть весьма эффективна в случае представления динамических данных, особенно тогда, когда у разработчика стоит цель быстрого поиска информации, в тех же базах данных, к примеру. Еще древом называют структуру данных, которая представляет собой совокупность элементов, а также отношений между этими элементами, что вместе образует иерархическую древовидную структуру.

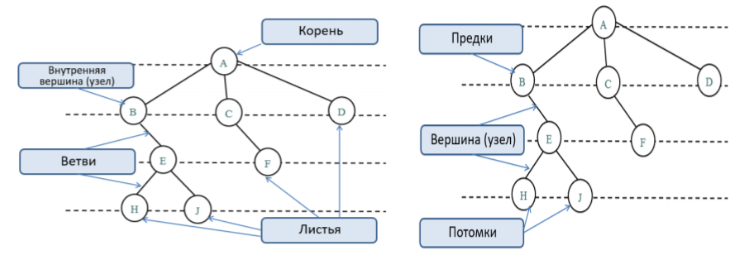
Дерево - это совокупность элементов, называемых узлами (один из которы определен как корень), и отношений ("родительских"), образующих иерархическую структуру узлов.

Рисунок 5

Чтобы выполнить конкретную операцию над всеми вершинами, надо все эти узлы просмотреть. Данную задачу называют обходом дерева. То есть обход представляет собой упорядоченную последовательность узлов, в которой каждый узел встречается лишь один раз.

В процессе обхода все узлы должны посещаться в некотором, заранее определенном порядке. Есть ряд способов обхода: прямой (префиксный, pre-order), симметричный (инфиксный, in-order), обратный (постфиксный, post-order).

Бинарные имеют степень не более двух. То есть двоичным древом можно назвать динамическую структуру данных, где каждый узел имеет не большое 2-х потомков. В результате двоичное дерево состоит из элементов, где каждый из элементов содержит информационное поле, а также не больше 2-х ссылок на различные поддеревья. На каждый элемент древа есть только одна ссылка.

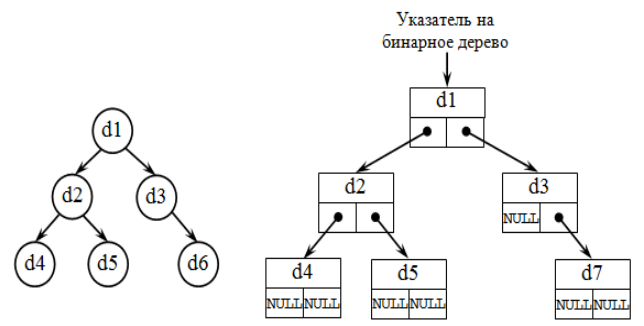
Самый удобный вид бинарного дерева – бинарное дерево поиска.

Рисунок 6

Деревья в программировании могут быть полезны в следующих случаях:

Когда программа должна обрабатывать генеалогическое древо либо работать со структурой каталогов. В таких ситуациях иногда есть смысл сохранять между объектами программы существующие иерархические отношения. В качестве примера можно вывести древо каталогов операционной системы UNIX (рис.7)

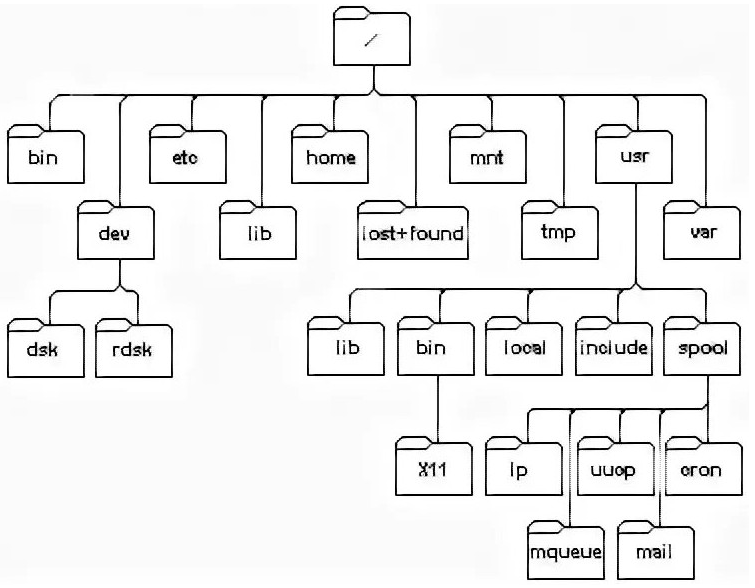


Рисунок 7

Когда между объектами, которые обрабатывает программа, отношения иерархии не заданы явно, но их можно задать, что сделает обработку данных удобнее (онлайн-переводчики с деревом синтаксического разбора).

Таким образом деревья используются в поисках баз данных, сортировке и выводе данных, вычислении арифметических выражений, кодировании по методу Хаффмана…

Кодирование Хаффмана может очень эффективно сжимать данные: обычно оно позволяет сэкономить от 20% до 90% пространства, а конкретная степень сжатия зависит от характеристик данных. Мы рассматриваем данные, подлежащие сжатию, как последовательность символов. В соответствии с частотой каждого символа жадный алгоритм Хаффмана строит оптимальное двоичное представление символа.

Кодирование Хаффмана - это тип кодирования с переменной длиной слова.

Хаффман разработал жадный алгоритм для построения оптимального префиксного кода, который называется кодом Хаффмана. Дерево Хаффмана: дерево Хаффмана также называется оптимальным двоичным деревом. Это двоичное дерево с наименьшей взвешенной длиной пути WPL среди всех двоичных деревьев, состоящих из n взвешенных листовых узлов.

# Тема 3.3 Основные виды деревьев

Дерево двоичного поиска — это структура данных, которая позволяет быстро работать с отсортированном списком чисел.

Чем отличается от обычного бинарного дерева

1. Все узлы левого поддерева меньше корневого узла.
2. Все узлы правого поддерева больше корневого узла.
3. Оба поддерева каждого узла тоже являются деревьями двоичного поиска, т. е. также обладают первыми двумя свойствами.

АВЛ-дерево — это прежде всего двоичное дерево поиска, ключи которого удовлетворяют стандартному свойству: ключ любого узла дерева не меньше любого ключа в левом поддереве данного узла и не больше любого ключа в правом поддереве этого узла.

Особенностью АВЛ-дерева является то, что оно является сбалансированным в следующем смысле: для любого узла дерева высота его правого поддерева отличается от высоты левого поддерева не более чем на единицу.

2-3 дерево — структура данных, представляющая собой сбалансированное дерево поиска, такое что из каждого узла может выходить две или три ветви и глубина всех листьев одинакова. Является частным случаем B+ дерева.

2-3 дерево — сбалансированное дерево поиска, обладающее следующими свойствами:

1. нелистовые вершины имеют либо 2, либо 3 сына,
2. нелистовая вершина, имеющая двух сыновей, хранит максимум левого поддерева. Нелистовая вершина, имеющая трех сыновей, хранит два значения. Первое значение хранит максимум левого поддерева, второе максимум центрального поддерева,
3. сыновья упорядочены по значению максимума поддерева сына,
4. все листья лежат на одной глубине,
5. высота 2-3 дерева O(log n), где n — количество элементов в дереве.

B-деревья также представляют собой сбалансированные деревья, поэтому время выполнения стандартных операций в них пропорционально высоте. Но, в отличие от остальных деревьев, они созданы специально для эффективной работы с дисковой памятью.

B-дерево — сильноветвящееся сбалансированное дерево поиска, позволяющее проводить поиск, добавление и удаление элементов за O(log n). B-дерево с n узлами имеет высоту O(log n). Количество детей узлов может быть от нескольких до тысяч (обычно степень ветвления B-дерева определяется характеристиками устройства (дисков), на котором производится работа с деревом).

Красно-чёрное дерево — один из видов самобалансирующихся двоичных деревьев поиска, гарантирующих логарифмический рост высоты дерева от числа узлов и позволяющее быстро выполнять основные операции дерева поиска: добавление, удаление и поиск узла. Сбалансированность достигается за счёт введения дополнительного атрибута узла дерева — «цвета». Этот атрибут может принимать одно из двух возможных значений — «чёрный» или «красный».

Свойства красно-черных деревьев:

1. Каждый узел окрашен либо в красный, либо в черный цвет (в структуре данных узла появляется дополнительное поле – бит цвета).
2. Корень окрашен в черный цвет.
3. Листья(так называемые NULL-узлы) окрашены в черный цвет.Каждый красный узел должен иметь два черных дочерних узла. Нужно отметить, что у черного узла могут быть черные дочерние узлы. Красные узлы в качестве дочерних могут иметь только черные.
4. Пути от узла к его листьям должны содержать одинаковое количество черных узлов(это черная высота).

# Список используемой литературы

1. «Алгоритмы и структуры данных» - О.Б.Фофанов - <https://portal.tpu.ru/SHARED/o/OFOFANO/work/Tab2/up.pdf>
2. Веб сайт OTUS JOURNAL – «Классификация структур данных» - <https://otus.ru/journal/klassifikaciya-struktur-dannyh/>
3. «Грокаем алгоритмы» - Адитья Бхаргава
4. «Основные структуры и алгоритмы по обработке информации» - <https://studbooks.net/2033938/informatika/strukturnost_dannyh_tehnologiya_programmirovaniya>
5. Prog-cpp – «Структуры данных» - <https://prog-cpp.ru/data-struct/>
6. «Введение в деревья» - <https://coderlessons.com/tutorials/akademicheskii/diskretnaia-matematika/vvedenie-v-derevia>
7. «Деревья выражений» - <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/expression-trees/>
8. <https://ppt-online.org/53214>
9. Веб сайт OTUS JOURNAL – «Дерево как структура данных - <https://otus.ru/journal/derevo-kak-struktura-dannyh/>
10. «Алгоритм Хаффмана» - <https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A5%D0%B0%D1%84%D1%84%D0%BC%D0%B0%D0%BD%D0%B0>
11. «Дерево двоичного поиска» - <https://codechick.io/tutorials/dsa/dsa-binary-search-tree>
12. «2-3 дерево» - <https://neerc.ifmo.ru/wiki/index.php?title=2-3_%D0%B4%D0%B5%D1%80%D0%B5%D0%B2%D0%BE>