**Структуры данных.**

**1. Постановка задачи.**

При разработке программ и алгоритмов важным этапом является этап подбора математической абстракции для описания данных, используемых в формулировке задачи. Например, в случае поиска оптимальной стратегии для игры чет-нечет таким объектом была игра, в случае задачи об Ариадне и Тезее - лабиринт, в задаче о ходе коня - шахматная доска, в примере из лекции 16 - учреждение. Будем называть предстваление этих объектов-данных ввиде математических абстракций Абстрактными Структурами Данных (АСД). В случае игры в качестве АСД мы использовали дерево; в случае лабиринта - граф; в случае шахматной доски - матрицу.

Выбрав подходящую по своим математическим свойствам структуру АСД, мы приходим к другой проблеме - как представить выбранную АСД в терминах тех структур данных, с которыми умеет работать исполнитель алгоритмов, которые есть в испоьзуемом языке программирования. Назовем эти струтктуры данных Структурами Данных Хранения (СДХ). Например, в случае задачи об Ариадне и Тезее мы представили граф, представляющий лабиринт, в виде матрицы смежности, которую мы представили в виде соотвествующей СДХ - массива, для шахматной доски мы применили ту же структуру данных для хранения данных задачи, для учреждения - мы использовали запись.

Критерием выбора для АСД подходящей СДХ является эффективность операций над СДХ, являющихся аналогами соотвествующих операций над АСД. Под эффективностью мы понимаем сложность алгоритмов над СДХ.

Итак, мы приходим к следующей проблеме: задано АСД, набор СДХ; требуется построить отображение АСД -> СДХ так, чтобы сложность алгоритмов операций над СДХ, соотвествующих надлежащим оперциям над АСД, была бы минимальной.

**2. Основные понятия и определения**

Структура G на множестве M - это пара (R,M) где R это отношение порядка на множестве M.

Определение.

Отношение порядка на множестве M это подмножество множества M\*M обладающее следующими свойствами:

1.a<=a - рефлективность;

2.если a<=b и b<=c, то a<=c - транзитивность;

3.если a<=b и b<=a, то a=b - антисимметричность.

Если отношение не обладает свойством антисимметричности, то оно называется предпорядком.

Определение.

Отношение порядка назывется линейным, если оно определено для любых a и b из M.

Определение.

Множество называется частично упорядоченным если на нем зафиксирован некоторый порядок.

Примеры.

1. Множество натуральных чисел с отношением <=.

2. Множество слов в алфавите с отношением лексико-графического упорядочения.

3. Множество людей с отношением родства.

4. Множество людей с отношением начальник-подчиненный.

Для выбора и обозачения элементов на M используют индексацию I: I - это отображение M -> [ 1.. |M| ].

**3. Абстрактные структуры данных.**

Наиболее часто встречающимися абстрактыми структурами данных являются строка, граф, дерево, стек, очередь, таблица.

**3.1. Строка**

Строка - конечное множество символов с отношением линейного прядка, определяющем следование символов в строке.

Примеры: текст, программа, формула.

Свойства строк:

1.Переменная длина;

2.Обращеие к элементам строки в отношении порядка, а не индексации;

3.Строка может иметь дополнительную структуру, называемую синтаксис, но это дополнительная структура.

Типичные операции:

поиск заданного символа;

вставка символа;

удаление символа;

замена заданного символа другим заданным символом.

**3.2. Граф**

Определение.

Графом G=(X,U) где X - множество, а U - отношение порядка на X.

Если U - отношение частичного порядка, то G ориентированный граф. Дело в том, что согласно определению частичного порядка из условия (a,b)принадл.U не следует (b,a)принадл.U.

Если U - предпорядок со свойством симметричности, то G - неориентированный.

Определение.

Граф G - взвешенный если задано отображение m:U -> R. Иногда вес называют длиной дуги.

Определение.

Граф G - размеченный (помеченный) если задано отображение W: X -> A (U->A), где A - множество меток.

Примеры.  
Объект Задача  
1. Сеть дорог Найти кратчайший маршрут  
2.Блок-схема программы Найти неиспользуемый участок  
3.Электрическая схема Вычислить характеристики цепи  
4.Общество Найти взаимосвязь групп  
5.Чертежи Сравнить  
6.Химические молекулы Поиск подструктур  
7.Сети ЭВМ Найти критический путь  
8.Поставка товаров Кто поставляет заданный товар

**Способы представления графов.**

Графический, матрица смежности, матрица инциденостей.

Определение.

Степенью вершины называется число дуг входящих и исходящих из этой вершины (инцидентных данной вершине).

Стпенью исхода (захода) вершины называется число вершин исходящих (входящих) в эту вершину.

Определение.

Граф называется регулярным если степень всех его вершин - величина постоянная.

Определение.

Последоватедность вершин (x1, x2, x3, ...xN ) графа G называется цепью если для любого i, принадл.[1..n] => {(x(i), x(i+1))принадл.U}.

Последоватедность вершин (x1, x2, x3, ...xN ) графа G называется путем если для любого i, принадл.[1..n] => {(x(i), x(i+1) )принадл.U и (x(i+1), x(i) )принадл.U}. Иными словами путь это цепь, рассматриваемая без учета ориентации вершин.

При этом n - длина пути, цепи соответственно.

Если x1=xN , то путь назывется циклом.

Если x1=xN в цепи, то цикл называется ориентированным.

Определение.

Граф назывется слабо связным если для любых его двух вершин есть путь их соединяющий без учета ориентации дуг графа.

Граф назывется сильно связным если для любых его двух вершин есть путь их соединяющий без с учетом ориентации дуг графа.

Определение.

Весом пути называется m(x1, x2, x3, ...xN) = m(x1, x2) + m(x2, x3) + ... + m(x(N-1), xN).

Типичные операции с графами:

вставить вершину;

удалить вершину;

заменить вершину;

добавить дугу;

удалить дугу;

найти в графе заданный подграф.

В терминах графов строка - линейный неориентированный граф.

Раздел математики изучающий свойства графов называется теорией графов.

**3.3.Деревья**

Определение.

Дерево - связный ациклический (без циклов) граф.

Подчеркнем что здесь в цикле ориентация дуг не учитывается.

Определение.

Одна вершина в дереве имеет степень захода 0. Эта вершина назывется корнем дерева.

Одна или несколько вершин в дереве имееют степень исхода 0. Эти вершины называются листьями.

Следствием условия ацикличности является то, что все вершины в графе, кроме корня, имеют сепень захода 1.

Деревья могут быть как ориентированные так и нет.

Определение.

Высотой дерева называется самый длинный путь в дереве из корня к листу.

Примеры.

1. Группа родственников с отношением быть ребенком;

2. Дерево игры;

3. Слова в алфавите: разбиваются на классы эквивалентности по первому символу, каждый класс в свою очередь разбивается на классы эквивалентности по второму символу и т.д.

Определение.

Не связный ациклический граф называется лесом.

Рекурсивное определение дерева.

1.Множество из одной вершины - дерево;

2.Если Т1, Т2, ...ТN - деревья, то вершина

T1 T2 T3 ... TN - дерево

Основные операции над деревьями:

найти заданную вершину;

заменить одну заданную вершину на другую;

удалить вершину;

вставить новую вершину.

При построении алгоритмов, реализующих эти равно как и другие операции над деревьями надо учитывать, что в дереве возможны самые разнообразные последователльности промотра вершин. Было бы полхо если результат операции зависел бы от порядка обхода дерева. Основные направления обхода: сверху-вниз, снизу-вверх, справа-налево, слева-направо.

Определение.

Дерево называется K-ичным если все внутренние вершины в нем имеют степень исхода K.

Соответсвенно дерево называется единичным или линейным, если степень исхода его внутренних вершин равна 1; в двоичном или бинарном дереве она соответсвенно равна 2.

Перечисление бинарных деревьев.

Попробуем подсчитать число двоичных деревьев, имеющих n вершин. Обознаячим b(i) - число бинарных деревьев из i вершин. Тогда следуя рекурсивному определению дерева, можно написать

b(n)=b(0)b(n-1) +b(1)b(n-2) +b(2)b(n-3) +...+b(n-1)b(0)

Нетрудно подсчитать что b(1)=1, b(2)=2, b(3)=b(0)b(2) + b(1)b(1) + b(2)b(0) , полагая b(0)=1, получаем, что b(3)=2+1+2=5. Соответственно b(4)=5+2+2+5=14, b(5)=14+5+4+5+14=42.

В общем случае b(n)=C(n, 2n) / (n+1) = (2\*n!) / (n! \* n!)

Определение.

Дерево называется совершенным (или полностью сбалансированным) если любой путь в нем от корня к листу не более чем на единицу отличается от длины самого длинного пути в этом дереве.

Следствие.

Совершенное дерево из n вершин имеет минимальную высоту среди всех деревьев, имеющих n вершин.

Подсчитаем высоту совершенного бинарного дерева из n вершин. Пусть в нем i ярусов. (Ярусом назовем множество вершин равно отстоящих от корня дерева.) Тогда число вершин в дереве можно выразить следующим соотношением:

1 + 2 + 2^2 +...+ 2^i = n.

Отсюда получаем 2^i -1=n, i=log2 (n+1) - 1. Или в приближенно log n. Отметим, что число совершенных деревьев составляет лишь малую часть общего числа деревьев. Например, при n=4 их всего 4.

Совершенные деревья интересны, например, тем что сложность доступа от корню к любому листу практически одна и та же. Здесь под сложностью доступа мы понимаем длину пути от корня к листу.

**3.4. Стек**

Стеком называется линейное дерево. В отличии от деревьев для операций со стеком есть ограничения. Доступ в стеке возможен только к корню дерева, которое в случае стека называется окном стека. Поэтому чтобы посмотреть элемент или изъять его или добавить новый - все это можно сделать только через корень. Примеры стека: подносы в столовой, патроны в обойме.

Основные операции: добавить элемент в стек, изъять элемент.

**3.5. Очередь**

Очередь - это линейное дерево, но в отличие от стека добавить элемент в очередь можно только в корень дерева, который в случае очереди называется хвостом или концом очереди, а изъять элемент из очереди можно только со стороны листа, который называется головой очереди.

Пример: банальная очередь в магазине, в столовой. Основные операции изъять элемент, добавить элемент.

**3.6. Таблица**

Упорядоченное множество пар (ключ,тело).

Примеры:

функция может быть представлена как пара (аргумент, результат),

таблица с записями о людях. В такой таблице ФИО - ключ, данные о человеке - тело. Мы здесь не будем подробно останавливаться на таблицах, так как им будет посвящено особое место в курсе.

**4. Структура данных хранения (СДХ)**

В Pascal можно выделить две базовые структуры хранения: вектор из записей и список. С идей списка мы уже сталкивались когда рассматривали динамические структуры данных. Когда мы не можем фиксировать заранее число компонентов в структуре. Однако, как мы увидим позднее связывание статических элемнтов памяти посредством ссылок в цепочки позволяет динамически упралять не только числом компонентов, но и структурой в целом. Например, когда мы заранее не знаем степень вершин в графе.

На базовый характер этих структур так же указывает и то, что в их терминах можно описать все остальные структуры данных в Pascal. Более этого, важно помнить и то, что "за кулисами Pascal" стоит ЭВМ с ее структурой памяти. которая есть вектор из слов - адресуемой единице памяти. памяти.