

**Написать краткий конспект лекции и ответить на контрольные вопросы.**

**Списки, графы, деревья и таблицы**

**Между данными, используемыми в той или иной информационной модели, всегда существуют некоторые связи, определяющие ту или иную структуру данных.**

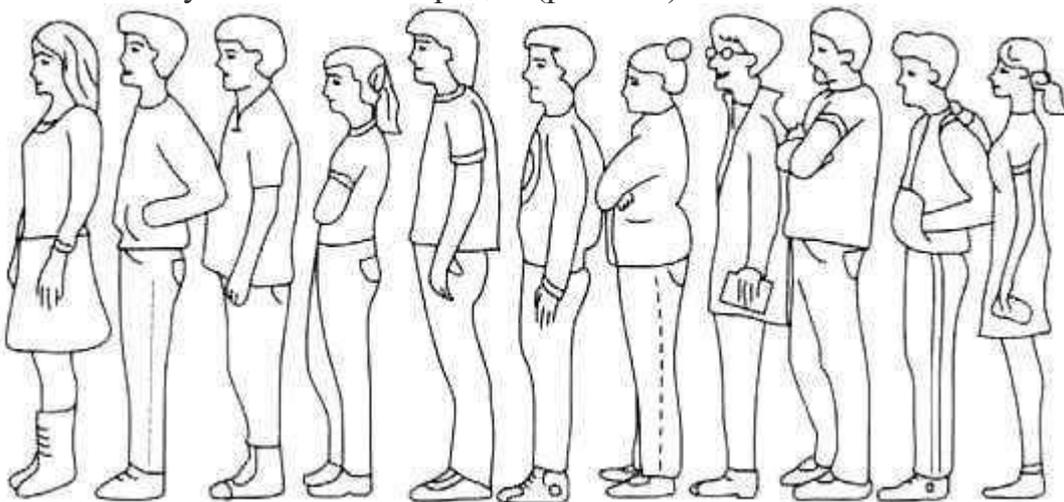
Вспомните, как мы определяли структуру данных при рассмотрении алгоритмов и программ. О каких информационных моделях тогда шла речь?

**Различают линейные и нелинейные структуры данных.**

В курсе информатики основной школы вы познакомились с линейным односвязным списком — последовательностью линейно связанных элементов, для которых разрешены операции добавления элемента в произвольное место списка и удаление любого элемента. Связь элементов списка осуществляется за счёт того, что каждый элемент списка содержит кроме данных адрес элемента, следующего за ним в списке. В линейном списке для каждого элемента, кроме первого, есть предыдущий элемент; для каждого элемента, кроме последнего, есть следующий элемент.

Частным случаем линейного односвязного списка является **стек** — **последовательность**, в которой включение и исключение элементов осуществляются с одной и той же стороны этой последовательности.

Ещё одним частным случаем линейного односвязного списка является **очередь** — **последовательность**, у которой включение элементов производится с одной стороны последовательности, а исключение — с другой. Сторона, где происходит включение элементов, называется хвостом; сторона, где происходит исключение, — головой. Понятие очереди как структуры данных очень близко к бытовому понятию «очередь» (рис. 3.2).



**Рис. 3.2. Иллюстрация понятия «очередь»**

Подумайте, какая связь между стеком и следующими объектами:

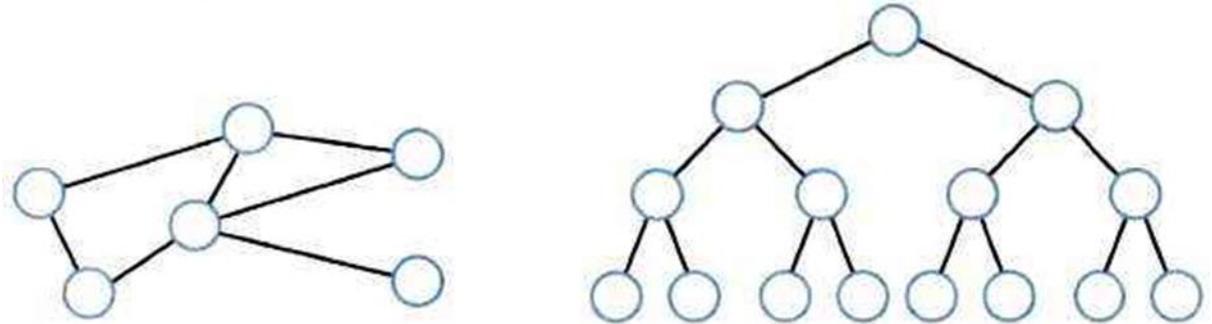


Почему стек является структурой типа **LIFO** (от англ. Last In, Firsts Out — последним пришёл, первым ушёл)?

Почему очередь является структурой типа **FIFO** (от англ. First In, First Out — первым пришёл, первым ушёл)?

Примеры нелинейных структур данных вам также хорошо известны — это **графы** и **деревья** (рис. 3.3).

**Граф** — это множество элементов (вершин графа) вместе с набором отношений между ними.



*Рис. 3.3. Примеры графовых структур*

**Графы** являются основным средством для описания структур сложных объектов. С их помощью можно описать вычислительную сеть, транспортную систему, схему авиалиний и другие объекты.

**Одной из разновидностей графа является дерево.**

**Дерево** — это совокупность элементов (вершин), в которой выделен один элемент (**корень**), а остальные элементы разбиты на **непересекающиеся множества (поддеревья)**. Каждое поддерево является деревом, а его корень является потомком корня дерева, т. е. все элементы связаны между собой отношением «предок — потомок». В результате образуется иерархическая структура вершин.

Частным случаем дерева является **бинарное дерево**, в котором каждая вершина может иметь не более двух потомков.

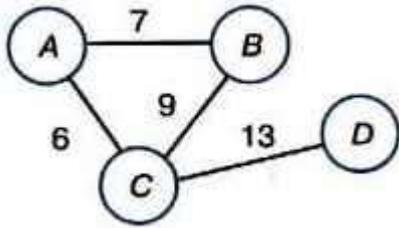
Деревья используются для представления родственных связей (генеалогическое дерево), для определения выигрышной стратегии в играх и т. д.

Ещё одной знакомой вам структурой данных являются таблицы, состоящие из строк и граф (столбцов, колонок), пересечение которых образуют ячейки. Таблицы применяют для наглядности и удобства сравнения показателей.

Табличный способ представления данных является универсальным — любую структуру данных, в том числе и представленную в форме графа, можно свести к

табличной форме. Это тем более важно в связи с тем, что для компьютерной обработки табличное представление данных является предпочтительным.

**Пример 1.** Построим таблицу, соответствующую неориентированному графу (рис. 3.5), отражающему схему дорог между некоторыми населёнными пунктами.



*Рис. 3.5. Граф схемы дорог*

Строки и столбцы таблицы будут соответствовать вершинам графа. Если две вершины являются смежными (соединены ребром), то в ячейку на пересечении соответствующих столбца и строки будем записывать вес этого ребра. В противном случае (вершины не являются смежными) в ячейку будем записывать 0. Получится таблица типа «объект — объект».

Такую таблицу называют матрицей смежности. Часто в матрицах смежности вместо нуля ставят знак минус, что обеспечивает большую наглядность.

	A	B	C	D
A	0	7	6	0
B	7	0	9	0
C	6	9	0	13
D	0	0	13	0

	A	B	C	D
A	-	7	6	-
B	7	-	9	-
C	6	9	-	13
D	-	-	13	-

Матрица смежности неориентированного графа симметрична относительно главной диагонали, идущей от левого верхнего угла к правому нижнему углу. У матрицы смежности неориентированного графа такая симметрия отсутствует.

**Пример 2.** Обед в школьной столовой состоит из двух блюд и напитка. На первое можно выбрать щи или окрошку, на второе — плов или пельмени, на третье — сок или компот. Все возможные варианты представлены с помощью дерева на рисунке 3.6.



*Рис. 3.6. Дерево вариантов обеда*

Для того чтобы представить эту же информацию в таблице, будем двигаться по дереву от листьев к корню, описывая все возможные варианты обеда.

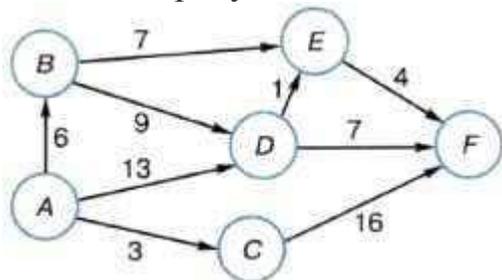
Обед	Напиток	2-е блюдо	1-е блюдо
Вариант 1	Сок	Плов	Щи
Вариант 2	Компот	Плов	Щи
Вариант 3	Сок	Пельмени	Щи
Вариант 4	Компот	Пельмени	Щи
Вариант 5	Компот	Плов	Окрошка
Вариант 6	Сок	Плов	Окрошка
Вариант 7	Компот	Пельмени	Окрошка
Вариант 8	Сок	Пельмени	Окрошка

Получилась таблица типа «объект-свойства»: объектами в ней являются варианты обеда, а свойствами — составляющие его блюда. При этом число граф в полученной таблице соответствует числу уровней в дереве.

При решении класса задач, связанного с нахождением кратчайшего пути в ориентированном графе, можно:

- 1) от исходного графа перейти к матрице смежности;
- 2) по матрице смежности построить дерево решений;
- 3) по дереву решений выбрать подходящий вариант.

**Пример 3.** Найдём кратчайший путь от вершины А до вершины F в графе, приведённом на рисунке 3.7.



**Рис. 3.7.** Ориентированный граф

Составим матрицу смежности, соответствующую данному ориентированному графу:

	A	B	C	D	E	F
A	-	6	3	13	-	-
B	-	-	-	9	7	-
C	-	-	-	-	-	16
D	-	-	-	-	1	7
E	-	-	-	-	-	4
F	-	-	-	-	-	-

3.8. По матрице смежности построим полное дерево перебора решений — рисунок

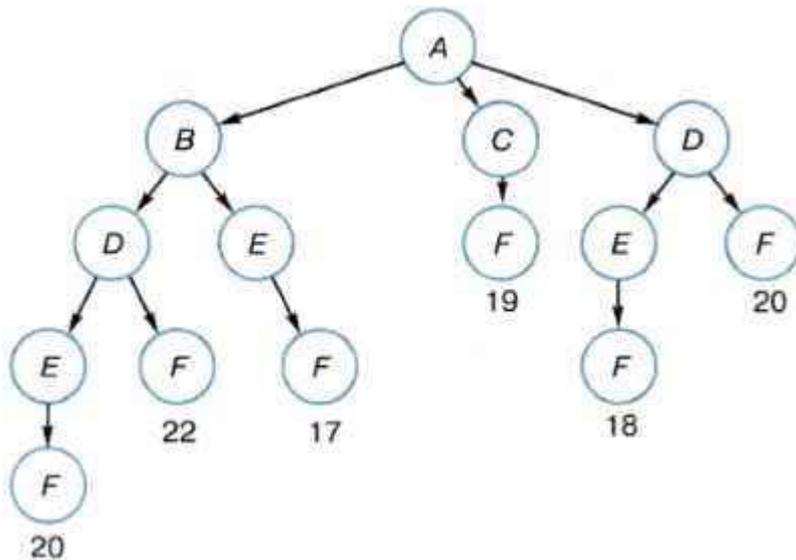
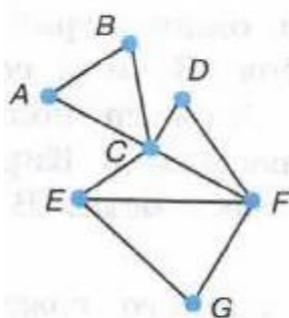


Рис. 3.8. Полное дерево перебора решений

На рисунке 3.8 видно, что кратчайший путь из вершины A в вершину F равен 17 и имеет вид A-B-E-F.

**Контрольные вопросы:**

- 1) Приведите примеры линейных структур данных. Чем очередь отличается от стека?
- 2) Почему графы и деревья считаются многоуровневыми структурами данных?
- 3) Что такое дерево? Какое дерево называется бинарным? Приведите примеры.
- 4) На рисунке представлена схема дорог, связывающих населённые пункты А, В, С, D, E, F, G. В таблице содержатся сведения о длинах этих дорог (в километрах). Схему и таблицу создавали независимо друг от друга, поэтому в них используются разные обозначения. Необходимо выяснить длину пути в километрах из пункта E в пункт F.



	Г1	Г2	Г3	Г4	Г5	Г6	Г7
Г1		9		2			
Г2	9			8		11	
Г3					3	12	
Г4	2	8				4	7
Г5			3			11	
Г6		11	12	4	11		9
Г7				7		9	