



- **ТЕМА 5.2 ВИДЫ ГРАФОВ**



Контрольные вопросы

1. Виды графов: простой, полный, псевдограф, мультиграф.
2. Эйлеров и гамильтонов графы.
3. Методика проверки графа на эйлеровость.
4. Деревья и их свойства.
5. Кодирование Пруффера для деревьев с пронумерованными вершинами



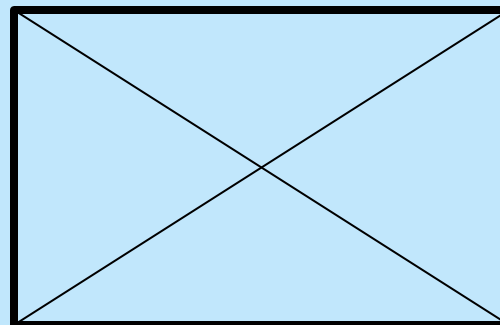
Виды графов

Неориентированный граф называется простым, если он не имеет петель и любая пара вершин соединена не более чем одним ребром.

Простой граф называется полным, если каждая пара вершин соединена ребром



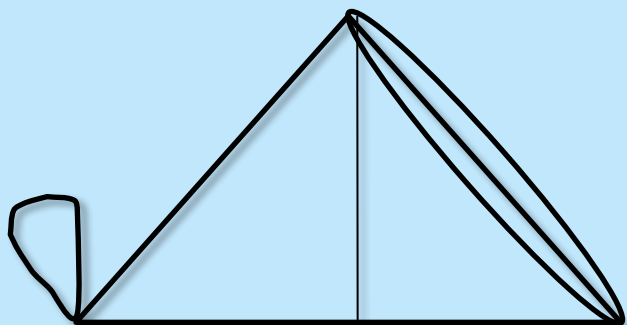
Простой граф



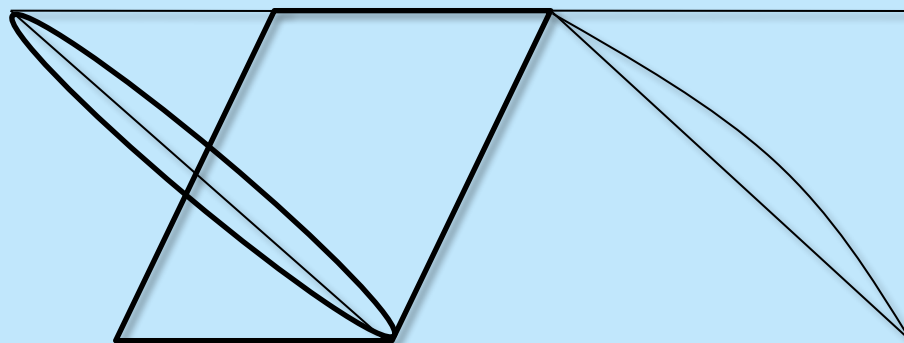
Полный граф



Псевдограф и мультиграф



псевдограф



мультиграф

Граф называется **псевдографом**, если в нем допускаются петли и кратные ребра, т.е. две вершины могут быть соединены более чем одним ребром.

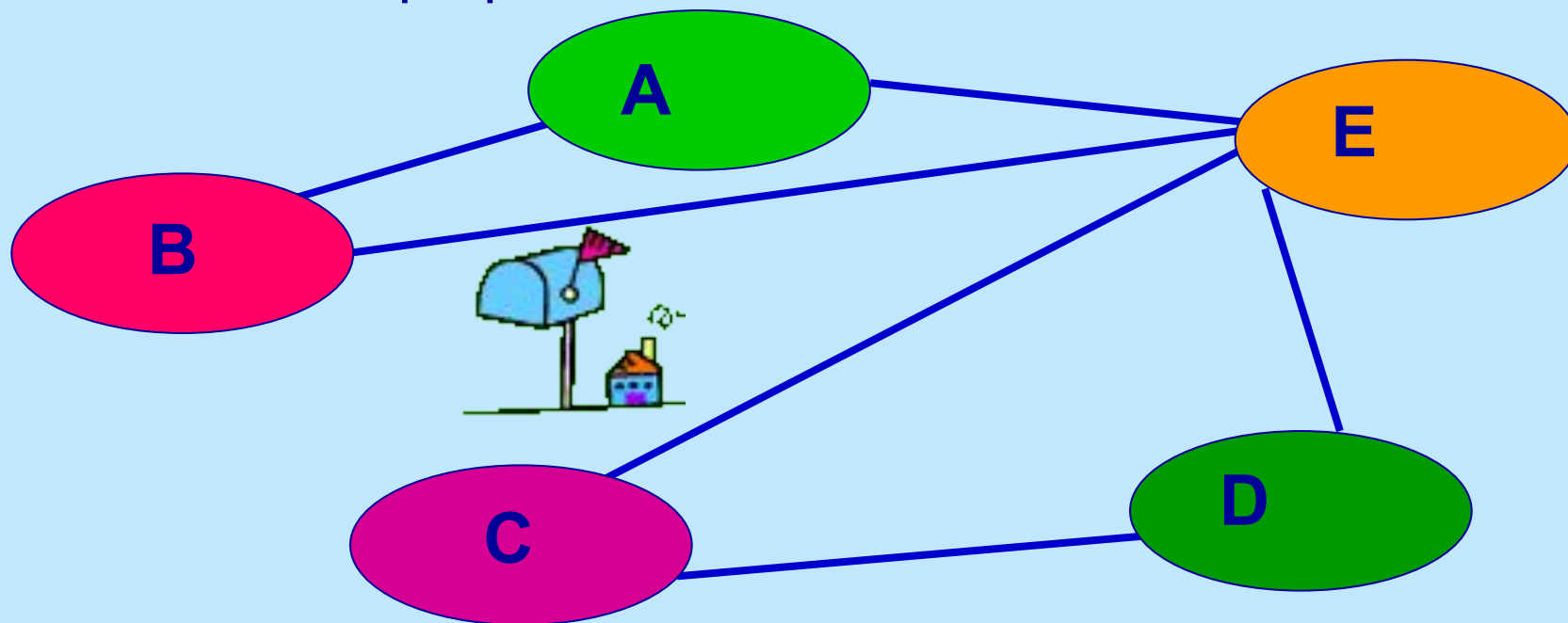
Псевдограф без петель называется **мультиграфом**.



Путь, цепь, цикл

Цепь – путь по вершинам и ребрам, включающий любое ребро графа не более одного раза.

Цикл – цепь, начальная и конечная вершины которой совпадают. Граф с циклом называют **сетью**.

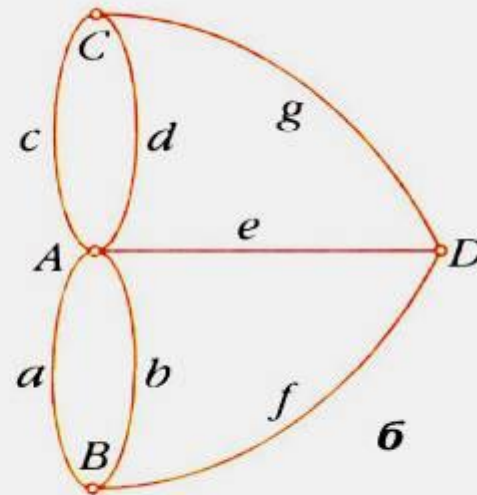
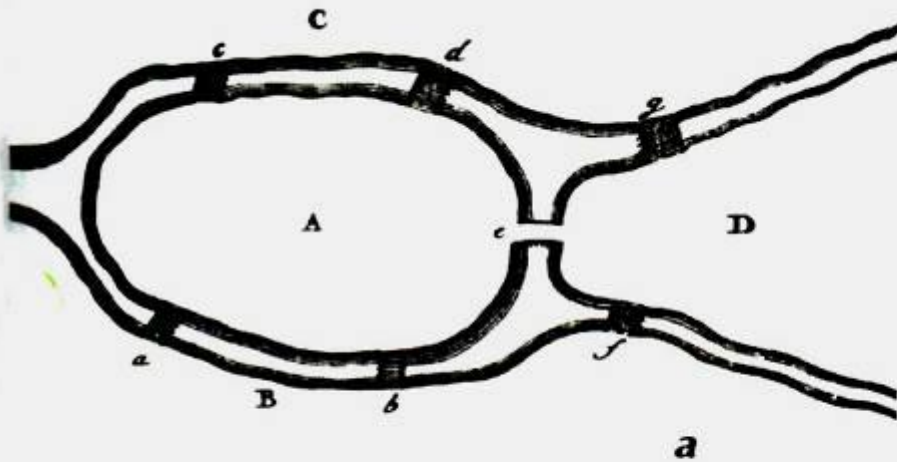


Приведите примеры цепи и цикла.



Эйлеровы графы

Бывший Кенигсберг, ныне Калининград, расположен на реке Прегель. А пределах города река омывает два острова. С берегов на острова были перекинуты мосты. Старые мосты не сохранились, но осталась карта города, где они изображены. Жители города предлагали приезжим следующую задачу: пройти по всем мостам и вернуться в начальный пункт, причём на каждом мосту следовало побывать только один раз.





Прогуляться по городским мостам предложили и Эйлеру. После безуспешной попытки совершить нужный обход, он начертил упрощённую схему мостов. Получился граф, вершины которого – части города, разделённые рекой, а рёбра – мосты. В итоге он доказал общее утверждение: **для того, чтобы можно было обойти всё рёбра графа по одному разу и вернуться в исходную вершину, необходимо и достаточно выполнение двух условий:**

Из любой вершины графа должен существовать путь по его рёбрам в любую другую вершину (граф должен быть связным);
Из каждой вершины должно выходить чётное количество рёбер.

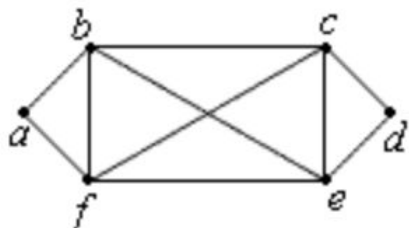


Методика нахождения эйлерова цикла в эйлеровом графе

1. Убедимся в том, что граф является эйлеровым (степени всех вершин четны).
2. Начиная с произвольной вершины, строим цепь, удаляя рёбра и запоминая вершины, пока не придём в ту же вершину, с которой начали.
3. Берём следующую вершину и продолжаем процесс, пока не будут удалены все рёбра.
4. Объединяем полученные последовательности в одну.

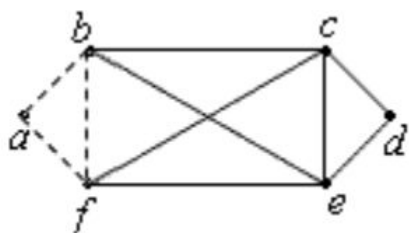


ПРИМЕР



$$d(a)=d(d)=2$$

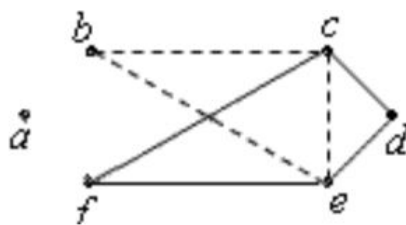
$d(b)=d(c)=d(e)=d(f)=4$ – граф эйлеров



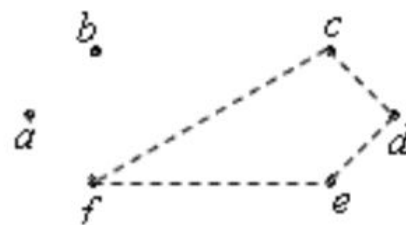
abfa

*ab**ce**bfa*

*ab**cd**ef**ce**bfa*



bceb



cdefc

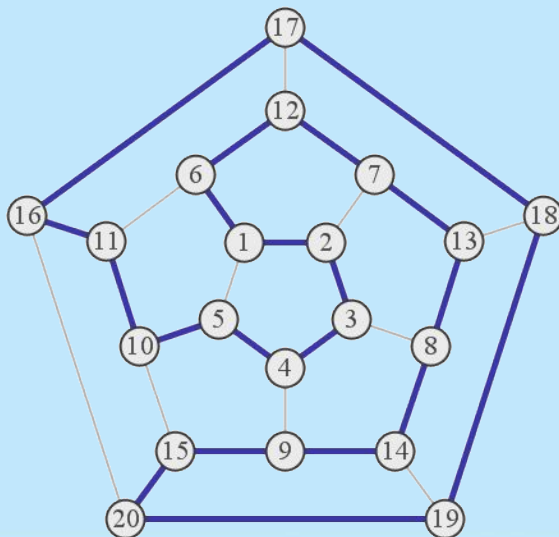


Гамильтоновы графы

Гамильтонов граф — граф, содержащий *гамильтонов цикл*.

При этом *гамильтоновым циклом* является такой цикл (замкнутый путь), который проходит через каждую вершину данного графа ровно по одному разу; то есть простой цикл, в который входят все вершины графа.

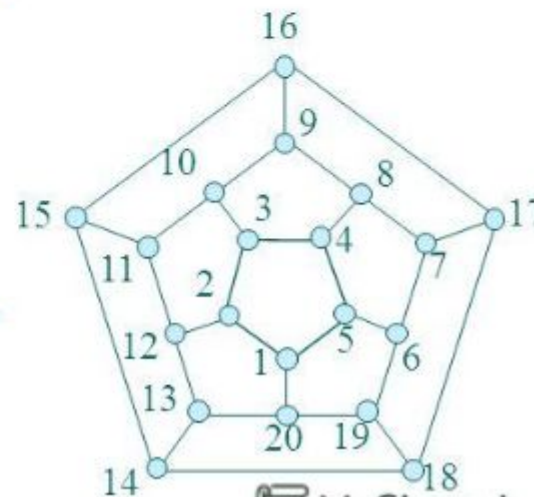
Также с гамильтоновым графом тесно связано понятие **гамильтонова пути**, который является простым путём (путём без петель), проходящим через каждую вершину графа ровно один раз. Гамильтонов путь отличается от цикла тем, что у пути начальные и конечные точки могут не совпадать, в отличие от цикла. Гамильтонов цикл является гамильтоновым путём.





Историческая справка

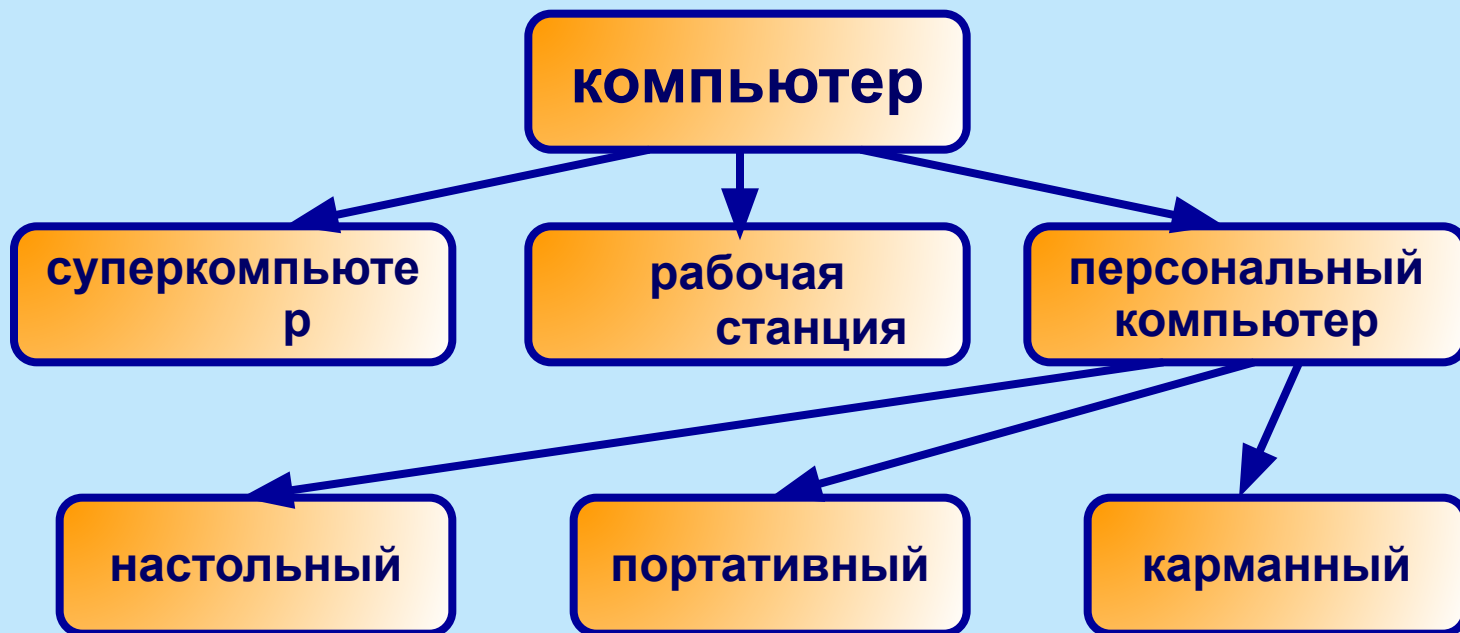
- Понятие гамильтонова цикла впервые появилось в связи с задачей о кругосветном путешествии, которую рассматривал Уильям Гамильтон: обойти все вершины графа – столицы различных стран – по одному разу и вернуться в исходный пункт
- Для 20 государств задача представляет обход всех вершин додекаэдра
- 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-1



В этой задаче вершины додекаэдра символизировали известные города, такие как [Брюссель](#), [Амстердам](#), [Эдинбург](#), [Пекин](#), [Прага](#), [Дели](#), [Франкфурт](#) и др., а рёбра — соединяющие их дороги.



Дерево – граф иерархической структуры. Между любыми двумя его вершинами существует единственный путь. Дерево не содержит циклов и петель.



Классификация компьютеров





Корень – главная вершина дерева.

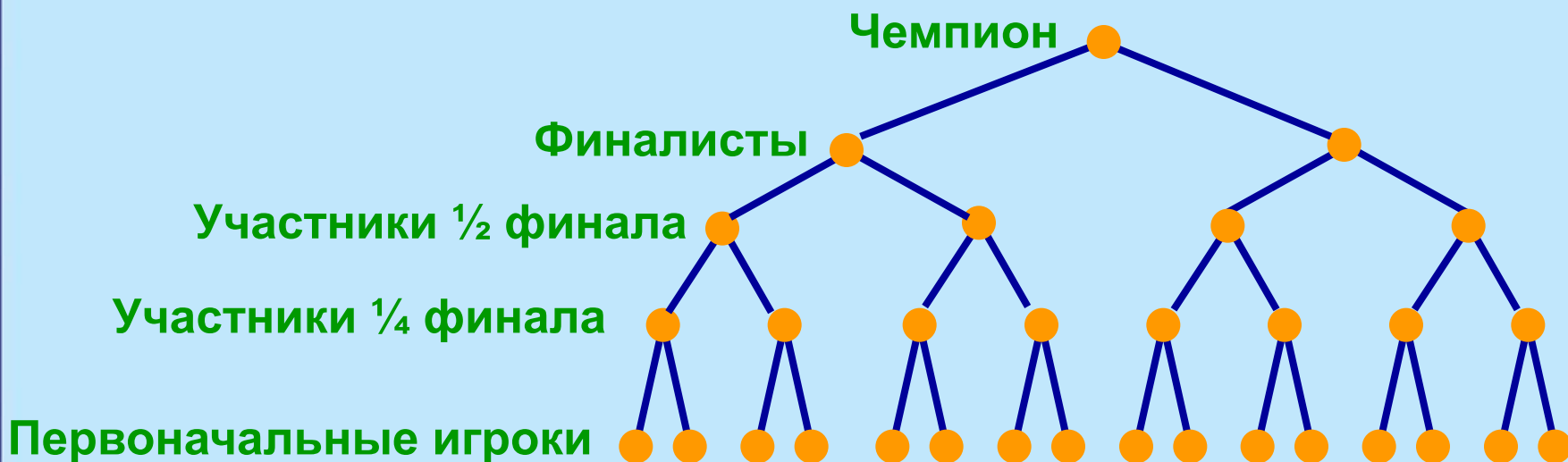
Предок – объект верхнего уровня.

Потомок – объект нижнего уровня.

Листья – вершины, не имеющие потомков.



Укажите перечисленные объекты у дерева

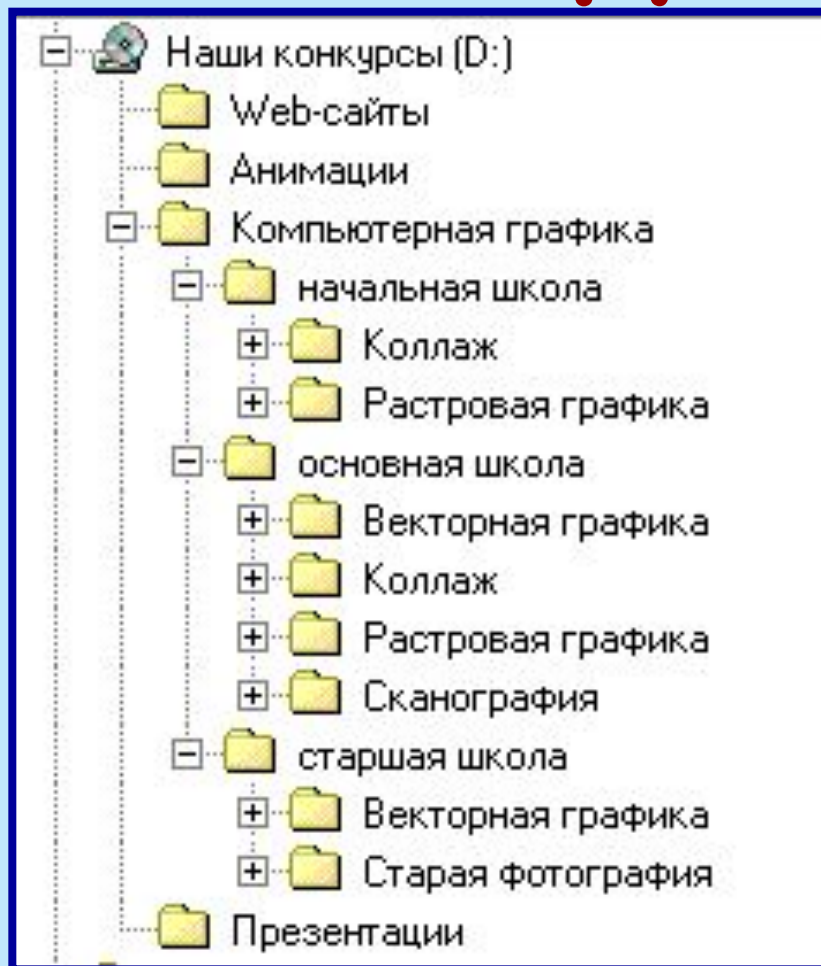


Олимпийская система спортивных соревнований





Файловая структура



Укажите корневую вершину, объекты 1-го, 2-го и 3-го уровней



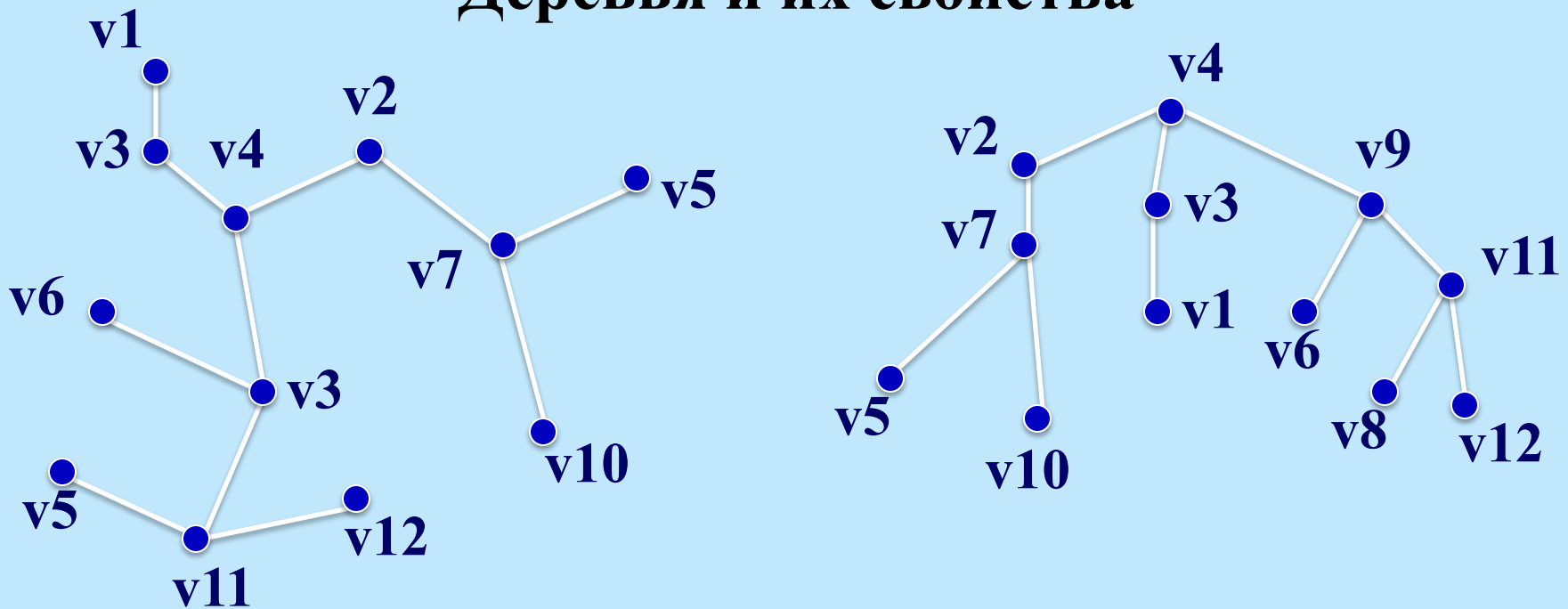


Самое главное

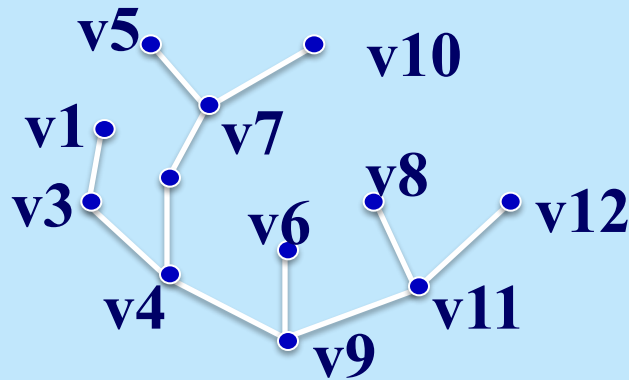
- **Граф** - наглядное средство представления состава и структуры системы. Граф состоит из вершин, связанных линиями. Направленная линия называется дугой, ненаправленная – ребром.
- **Иерархия** - расположение частей (элементов) целого в порядке от высшего к низшему. Системы, элементы которых находятся в отношениях подчиненности, называются иерархическими системами.
- **Дерево** - граф иерархической системы. Между любыми двумя вершинами дерева существует единственный путь.



Деревья и их свойства



Деревья являются простым, но очень важным видом графа. С их помощью описывается структура самых различных объектов: организации и учреждения, книг и документов, математических формул, компьютерных файловых систем, программ...



Листом (висячей вершиной) называют вершину степень которой равна 1, если она не рассматривается как корень. В качестве корня в неориентированном дереве можно принять любую вершину

Основные свойства деревьев:

1. дерево – это связный граф без циклов
2. любая пара вершин в дереве связана единственной цепью
3. дерево – это связный граф, имеющий n -вершин и $n-1$ ребер

Естественным понятием расширения дерева является лес – несвязный граф, все компоненты которого деревья



Кодирование Пруффера для деревьев с пронумерованными вершинами

Каждое дерево может быть однозначно записано числовым кодом, содержащим $n-2$ элемента и называемым кодом Пруффера

Алгоритм получения кода

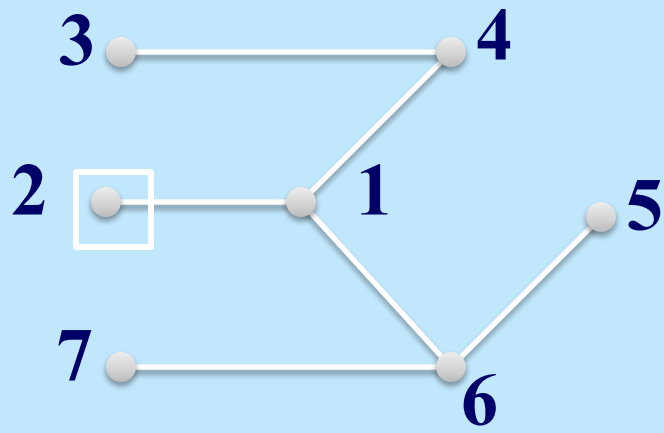
Пусть вершины дерева пронумерованы числами от 1 до n . Отыскиваем вершины степени 1 с наименьшим номером и включаем в ход номер смежный с ней вершиной, после чего удаляем найденную вершину вместе с ребром.

С получившимся подграфом выполняем ту же операцию повторяя её, пока только останется два ребра

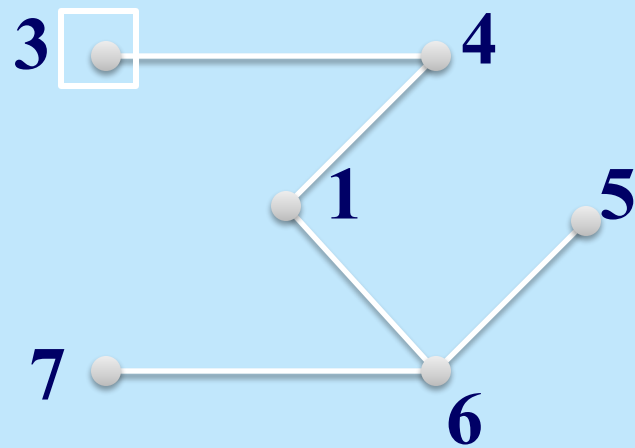


пример:

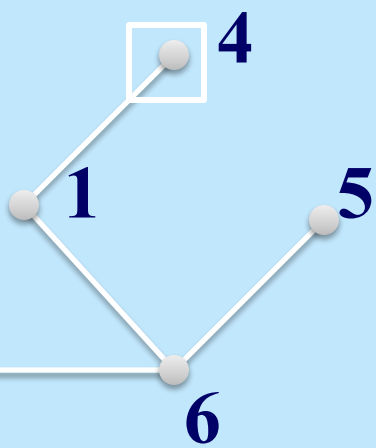
14166



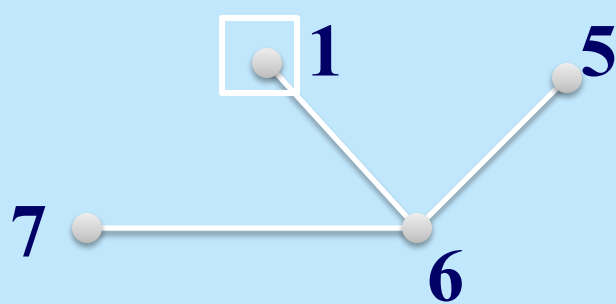
Код - 1



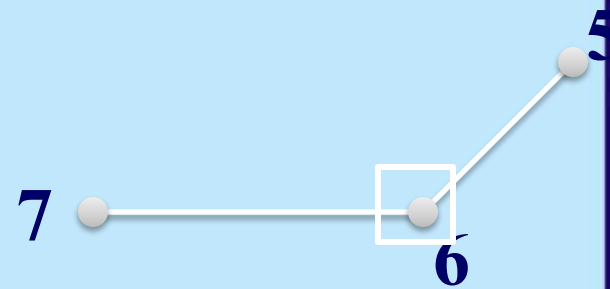
Код - 4



Код - 1



Код - 6



Код - 6



Восстановление дерева по коду

Антикод – упорядоченное по возрастанию последовательность номеров вершин не вошедших в код Пруффера

Для примера 2, 3, 5, 7

Дерево строится путем добавления ребер. Очередное добавляемое ребро начиная с первого образуется парой вершин номера которых стоят 1 в строке кода и в строке антикода. После этого использованные элементы строк вычеркиваются. Если номер вычеркнутый из строки кода не содержится среди оставшихся в ней элементов его следует добавить в строке антикода не нарушая её упорядоченность. Такие же действия повторяются с остатками строк кода и антикода. Пока они будут вычеркнуты все элементы первой из них



При этом строка антикода будет содержать 2 элемента образующих последнее ребро, которое следует добавить к формируемому дереву

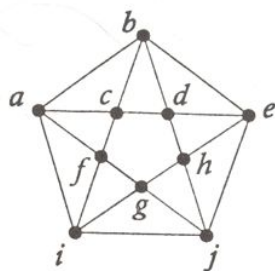
Итерация	1	2	3	4	5	6	7	8	9	10	11
Строка кода	1	4	1	6	6						
Строка антитеза	2	3	5	7							
		3	5	7							
			4	5	7						
				1	5	7					
					5	7					
						6	7				
Добавляемое ребро	(1;2)	(4;3)	(1;4)	(6;1)	(6;5)	(6;7)					



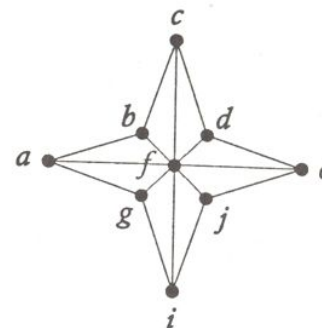
Домашнее задание

1. Определите является ли граф эйлеровым?

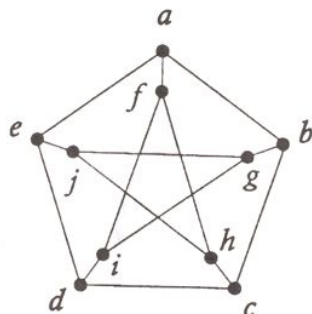
а)



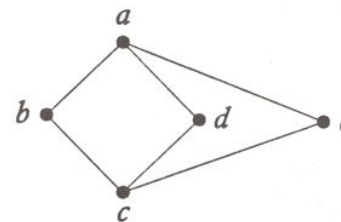
б)



в)



г)





Домашнее задание

2. Запишите код Пруффера для дерева.

