ГРАФЫ

Граф – это совокупность двух конечных множеств: множества точек и множества линий, попарно соединяющих некоторые из этих точек.

Множество точек называется ***вершинами (узлами)*** *графа*.

Множество линий, соединяющих вершины графа, называются ***ребрами (дугами)*** *графа*.

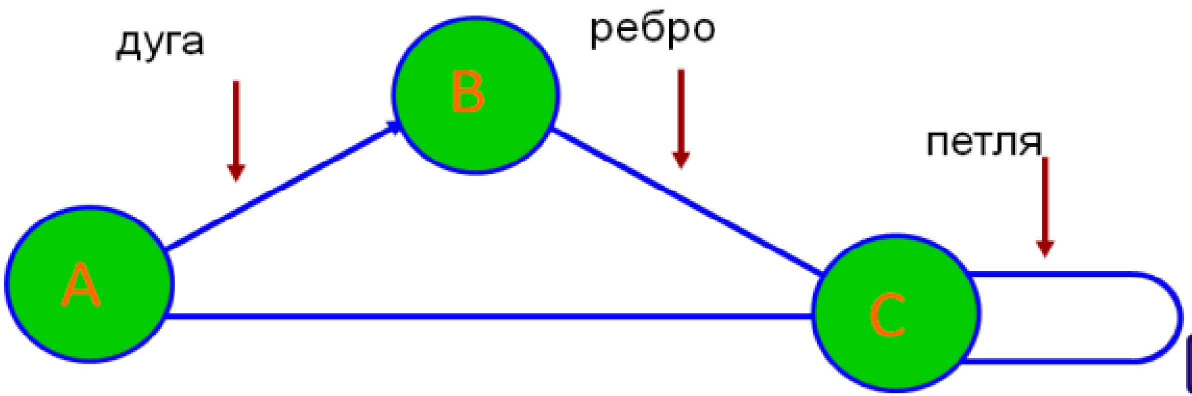
Состав графа

Граф состоит из ***вершин***, связанных линиями.

Направленная линия (со стрелкой) называется ***дугой***.

Линия ненаправленная (без стрелки) называется ***ребром***.

Линия, выходящая из некоторой вершины и входящая в неё же, называется ***петлей***.



- ГРАФ - пара **G = < V,E >,**
- где **V** - конечное непустое множество вершин,
- **E** - множество ребер (пар вершин).
- Если парам E не присвоены направления - граф **неориентированный**, иначе - **ориентированный**.
- *Смешанный граф* – граф, содержащий как ориентированные, так и неориентированные ребра.
- *Петля* - ребро, соединяющее вершину саму с собой. Если в пары E входят только различные вершины - **граф без петель.**
- Две вершины называются *смежными*, если существует соединяющее их ребро.
- Ребра, соединяющие одну и ту же пару вершин, называются *кратными*.

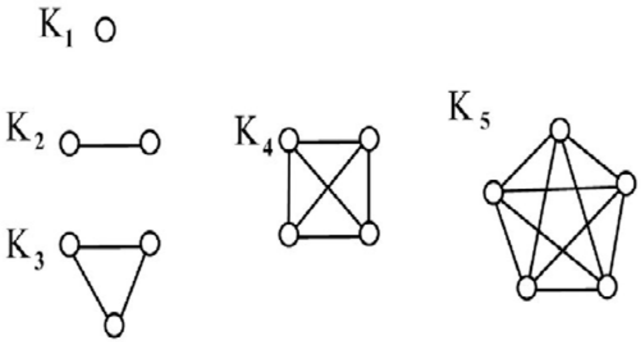
- **Простой граф** не имеет ни петель, ни кратных ребер.
- **Мультиграф** – это граф, у которого любые две вершины соединены более чем одним ребром.
- **Маршрут** в графе - это конечная последовательность смежных вершин и ребер, соединяющих эти вершины.
- Маршрут ***открытый***, если его начальная и конечная вершины различны, иначе ***замкнутый***
- **Цепь –** маршрут со всеми ***различными*** ребрами.
- **Путь –** открытая цепь с различными вершинами.
- **Полным графом** называется граф, в котором каждая вершина соединена ребром с любой другой вершиной

Мультиграф

Полный граф K_n

Граф полный, если каждая вершина смежна с каждой.

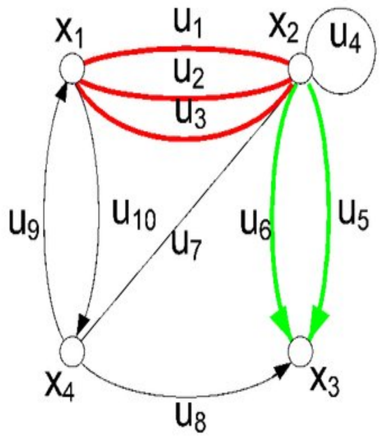
Полный граф с n вершинами - K_n



Граф, у которого хотя бы для *двух ребер* $u_j, u_l \in U$ справедливо

$$\Gamma_2 u_j = \Gamma_2 u_l \,\&\, \Gamma_1 u_j = \Gamma_1 u_l,$$

называется *мультиграфом*, а максимальное количество кратных ребер – *мультичислом*.



Мультичисло:

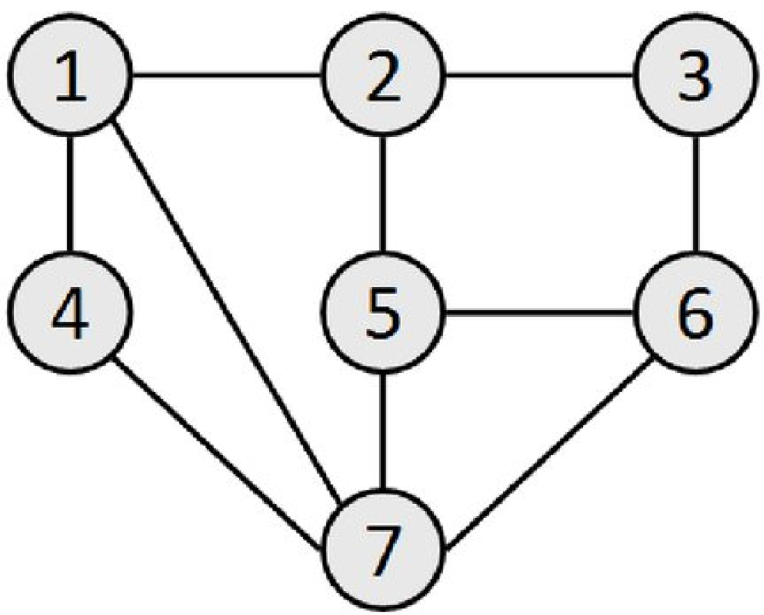
неориентированных ребер r = 3,

ориентированных – r = 2

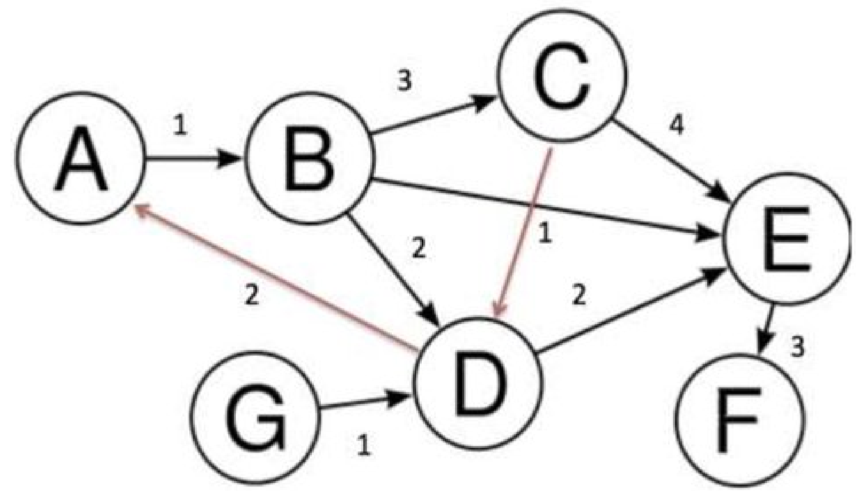
- **Цикл** - замкнутая цепь, если различны все ее вершины, за исключением концевых.
- Граф называется ***связным***, если для любой пары вершин существует соединяющий их путь.
- **Вес вершины** – число (действительное, целое или рациональное), поставленное в соответствие данной вершине (стоимость, пропускная способность и т. д.).
- **Вес (длина)** *ребра* – число или несколько чисел, которые интерпретируются по отношению к ребру как длина, пропускная способность и т. д.
- **Взвешенный** *граф* – граф, все ребра которого имеют вес.
- **Степень вершины** графа – это число ребер, инцидентных данной вершине, причем петли учитываются дважды. Поскольку каждое ребро инцидентно двум вершинам, сумма степеней всех вершин графа равна удвоенному количеству ребер: Sum(deg(vi), i=1…|V|) = 2 * |E|.

Графы: примеры

Неориентированный невзвешенный



Ориентированный взвешенный



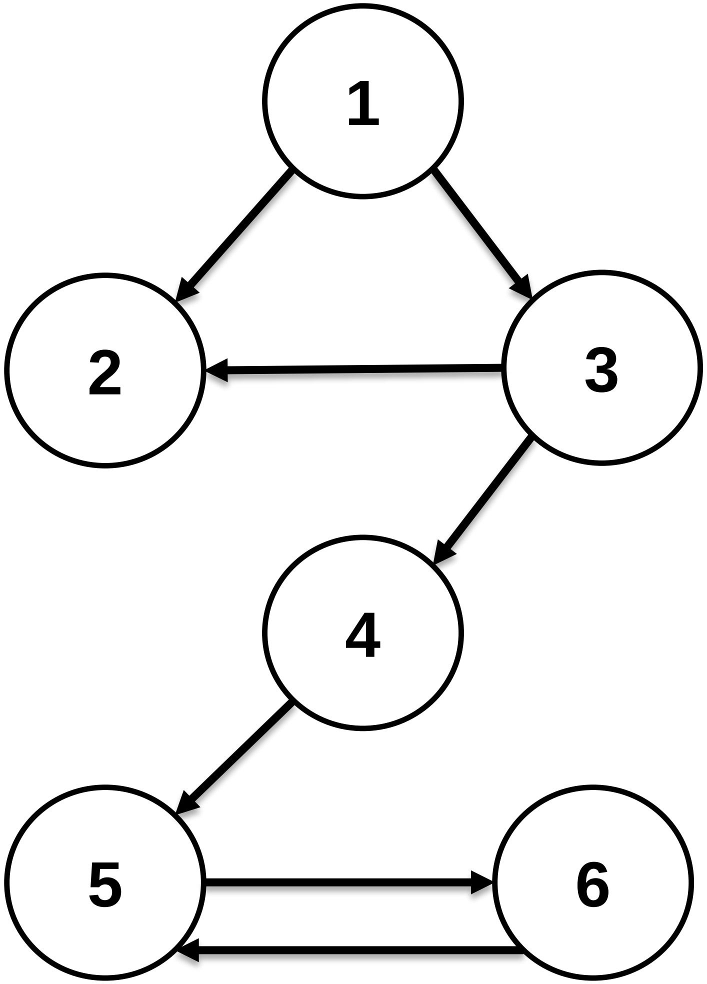
ПРЕДСТАВЛЕНИЕ ГРАФОВ

Количество вершин **n = ¦ V ¦,**
количество **ребер m = ¦ E ¦.**

- В математике классический способ представления графа – **МАТРИЦА ИНЦИДЕНЦИЙ A (n x m)**.

n строк - соответствующих вершинам
m столбцов - соответствующих ребрам.
Для орграфа столбец, соответствующий ребру **< X,Y >** из **|E|,** содержит
1 в строке, соответствующей вершине **X**;
-1 в строке, соответствующей вершине **Y**;
0 -в остальных строках.
В случае петли (ребро типа <X,X>)
возможна постановка другого числа (напр., 2).

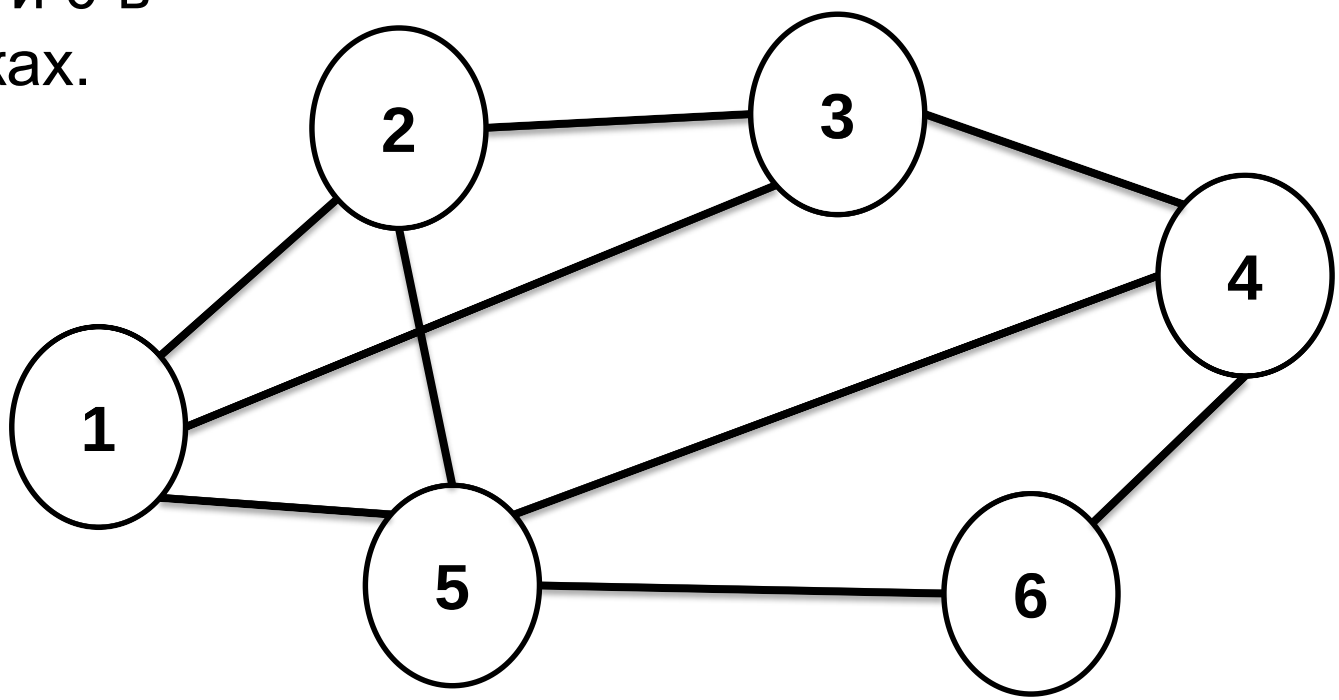
МАТРИЦА ИНЦИДЕНТНОСТИ для ориентированного графа



	1,2	1,3	3,2	3,4	4,5	5,6	6,5
1	1	1	0	0	0	0	0
2	-1	0	-1	0	0	0	0
3	0	-1	1	1	0	0	0
4	0	0	0	-1	1	0	0
5	0	0	0	0	-1	1	-1
6	0	0	0	0	0	-1	1

Для неориентированного графа столбец, соответствующий ребру **<X,Y>** содержит 1 в строках, соответствующих вершинам **X** и **Y** и 0 в остальных строках.

	1,2	**1,3**	**1,5**	**2,3**	**2,5**	**3,4**	**4,5**	**4,6**	**5,6**
1	1	1	1	0	0	0	0	0	0
2	1	0	0	1	1	0	0	0	0
3	0	1	0	1	0	1	0	0	0
4	0	0	0	0	0	1	1	1	0
5	0	0	1	0	1	0	1	0	1
6	0	0	0	0	0	0	0	1	1



матрица инцидентности

- Для представления графа матрицей инцидентности надо n x m элементов информации (ячеек памяти), из кот. большинство - нули. Неудобен и доступ к информации. Например, в пределе требуется перебор всех столбцов матрицы
- т.е. m шагов для ответа на вопросы типа "к каким вершинам ведут ребра из x?" или "существует ли ребро < x,y >?".
- Матрица инцидентности лучше всего подходит для операции «перечисление ребер, инцидентных вершине x»

МАТРИЦА СМЕЖНОСТИ

Более эффективно представление графа

МАТРИЦЕЙ СМЕЖНОСТИ B(N * N);

в ней элемент b[x,y] = 1 (или значение веса ребра), если есть ребро из вершины X в вершину Y и b[x,y] = 0 в противоположном случае. У неориентированных графов ребро {X,Y} является ребром {Y,X}, а потому матрица смежности для них всегда симметрична.

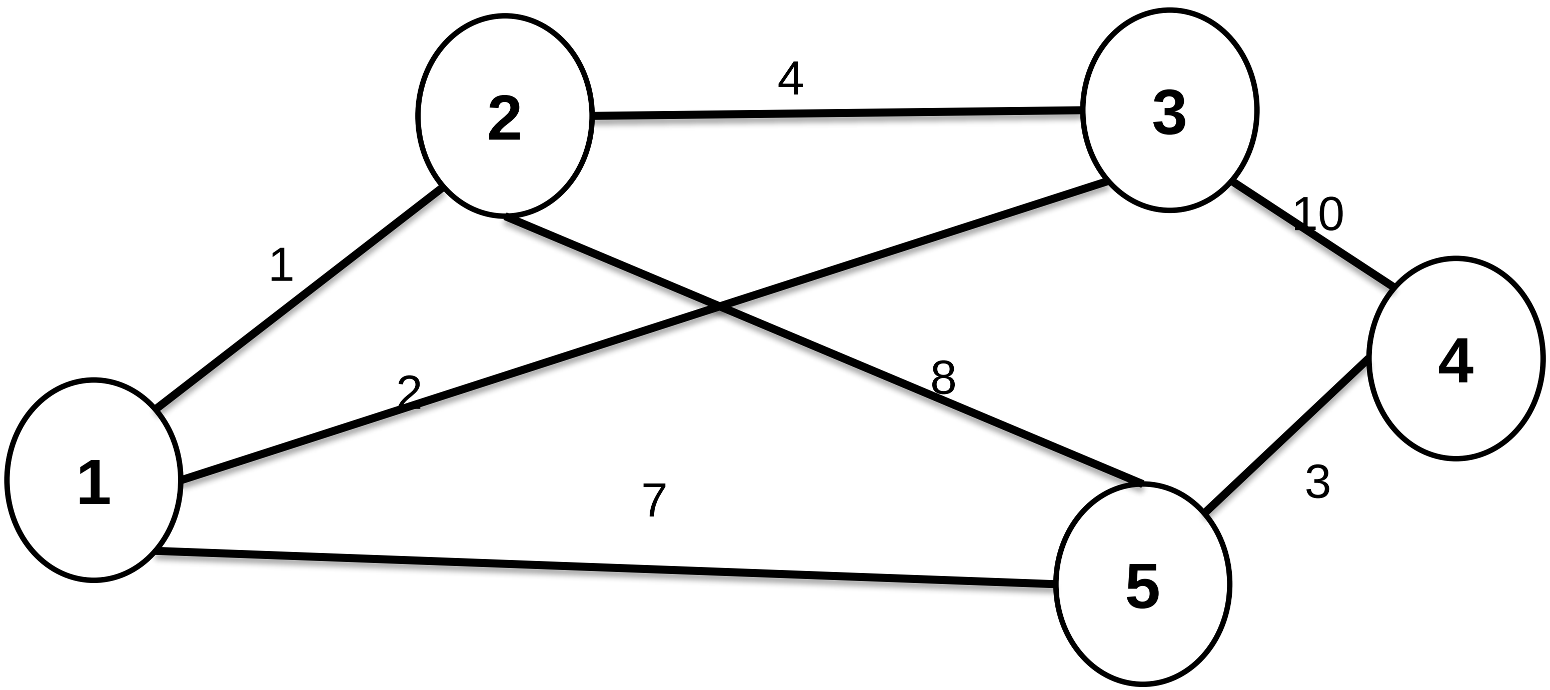
\+ простота

\- разреженность, нельзя представить несколько ребер между вершинами.

Этот способ очень хорош, когда нам надо часто проверять смежность или находить вес ребра по двум заданным вершинам.

Матрица смежности

	1	**2**	**3**	**4**	**5**
1	0	1	2	0	7
2	1	0	4	0	8
3	2	4	0	10	0
4	0	0	10	0	3
5	7	8	0	3	0

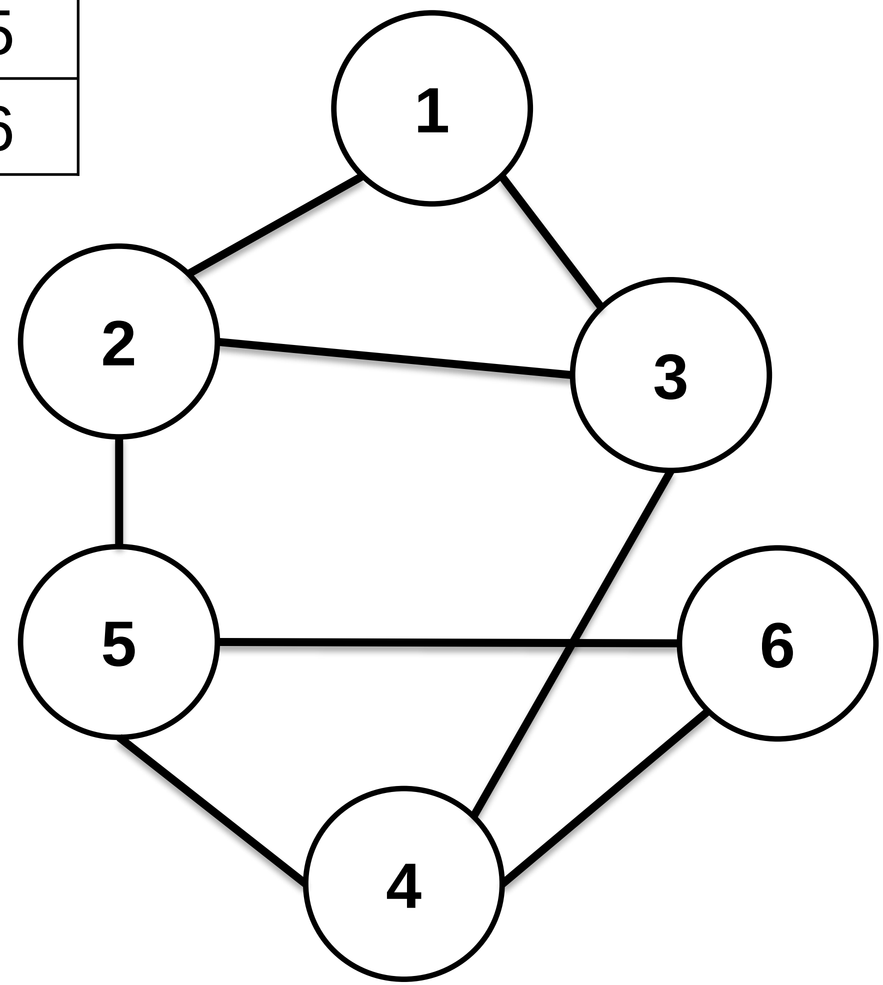


- Для уменьшения времени доступа к эл-ту матрицы можно использовать векторы Айлиффа.
- В этом случае за один шаг просмотра можно ответить на вопрос, существует ли ребро из X в Y. Однако, независимо от количества ребер, объем требуемой памяти равен **N*N**, хотя для малых **n** можно хранить строку (столбец) матрицы в машинном слове.

Список ребер

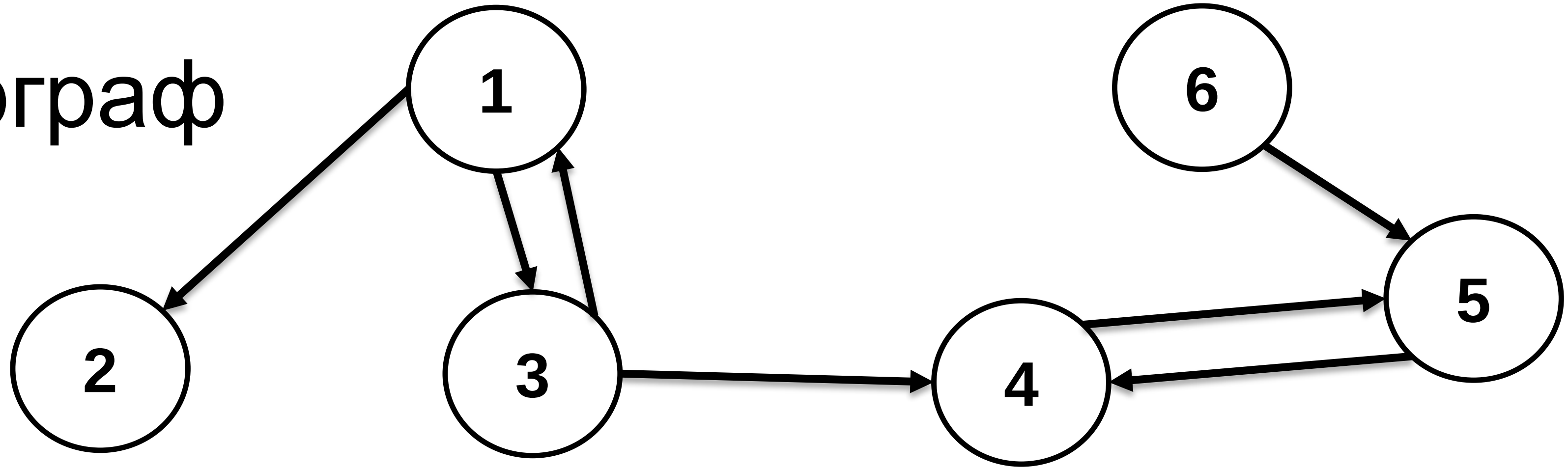
- Можно представить неограф списком ребер. Это экономит память при неплотных графах (m<<n*n).

1	1	2	2	3	4	4	5
2	3	3	5	4	5	6	6



В третьей строке таблицы при необходимости можно указывать вес ребра

Орграф



1	1	3	3	5	4	6
2	3	1	4	4	5	5

- Требуемый объем памяти: 2*m. Для взвешенного графа: 3*m (m - количество ребер).
- Неудобна; но, затрата, в худшем случае, порядка **m** шагов для получения множества вершин, непосредственно связанных с данной вершиной. Затраты можно значительно уменьшить, лексикографически упорядочив множество пар и применив двоичный поиск.

СПИСОК ИНЦИДЕНТНОСТИ

Для каждой вершины v из V содержит список вершин U, таких, что **v --> u** (для неориентированного графа v -- u).

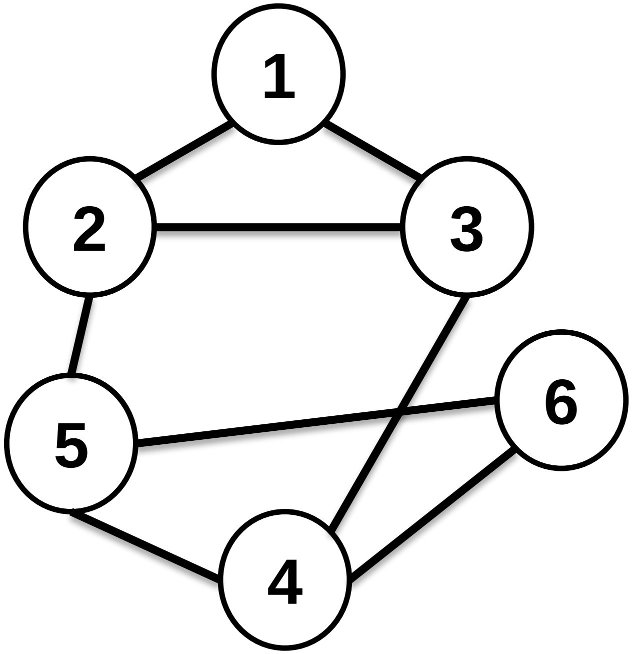
Каждый элемент списка инцидентности является записью (структурой)

R =Record

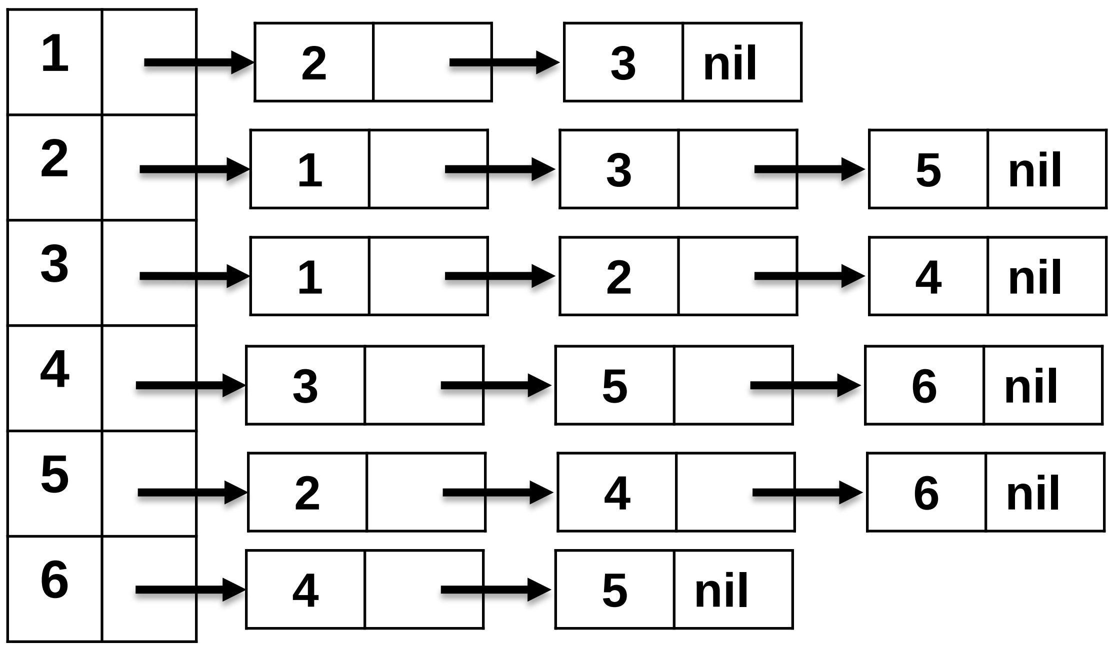
LINE : TypeV; //вершина

NEXT :^R; //указатель на следующую в списке

End; //(для последней записи в списке **.**NEXT = nil).



Для неориентированных графов каждое ребро **{u,v}** представлено дважды: через вершину v в списке **ZAP[u]** и через вершину u в списке **ZAP[v].** Требуемый объем памяти для представления графа списками инцидентности имеет порядок **m+n**.



- Начало каждого списка хранится в таблице **BEG**.

Т.е. **BEG[v]** - указатель на начало списка, содержащего вершины, непосредственно связанные с вершиной **v**; **{ u: v --> u},**

а для неориентированного графа **{ u: v -- u }.** Весь список такого вида обозначим **ZAP[v],**

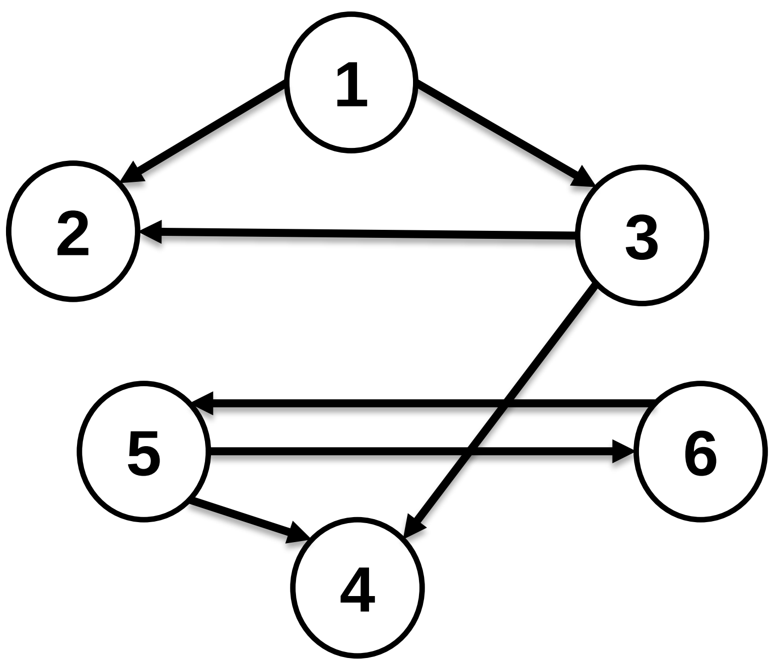
а организацию цикла, выполняющего определенную операцию для каждого элемента **u** из такого списка в последовательности по очереди элементов в списке, будем записывать в виде: **ДЛЯ u ИЗ** ZAP[v] **ДЕЛАТЬ**...

или: **For u FROM** ZAP[v] **Do …**

Для мало меняющихся графов м. исп-ть курсоры на массив смежных вершин:

1	2	3	4	5	6

2	3	0	2	4	0	4	6	0	5	0	



вершины графа пронумерованы от 1 до n, ребра – от 1 до m

	Способы представления графа			
Операция	**Матрица смежности**	**Матрица инцидентности**	**Списки смежных вершин**	**Список ребер**
	Емкостная сложность: $O(n^2)$	Емкостная сложность: $O(n*m)$	Емкостная сложность: $O(n^2)$	**Емкостная сложность: O(m)**
	Временная сложность			
Проверка смежности вершин X и Y	O (1)	O(n*m)	O (n)	O (m)
Перечисление всех вершин, смежных с X	O (n)	O(n*m)	O (n)	O (m)
Определение веса ребра {X, Y}	O (1)	O(n*m)	O (n)	O (m)
Определение веса вершины X	O (1)	Вес не хранится	Вес не хранится	Вес не хранится
Перечисление всех ребер {X, Y}	$O(n^2)$	O (m)	O (m)	O (m)
Перечисление ребер, инцидентных вершине X	Номера ребер не хранятся	O (m)	№ ребер не хранится	O (m)
Перечисление вершин, инцидентных ребру S	Номера ребер не хранятся	O (n)	№ ребер не хранится	O (1)

Основные процедуры обработки графовых структур :

- поиск вершин в графе
- поиск кратчайших путей от Vk до Vm
- поиск Эйлерова пути
- поиск Гамильтонова пути
- поиск кратчайших путей между всеми вершинами