3.9 Деревья

Дерево – это связный граф без циклов.

Свойства дерева:

- Любые две вершины в дереве связаны единственной простой цепью
- При удалении любого ребра дерева нарушается его связность.
- Число вершин в дереве на единицу больше числа ребер.

Ориентированные (упорядоченные) деревья обладают следующими дополнительными свойствами:

- Существует единственный узел, у которого полустепень захода равна нулю – корень дерева.
- Полустепень захода всех остальных узлов равна 1.
- Каждый узел достижим из корня.
- Узлы дерева с нулевой полустепенью исхода принято называть листьями.

3.10 Способы представления графа

Уточнение: число вершин графа обозначаем через *n*, а число ребер – через *m*.

Характеристика *M(n,m)*, приведенная для каждого способа представления графа, означает требуемый для него объем памяти.

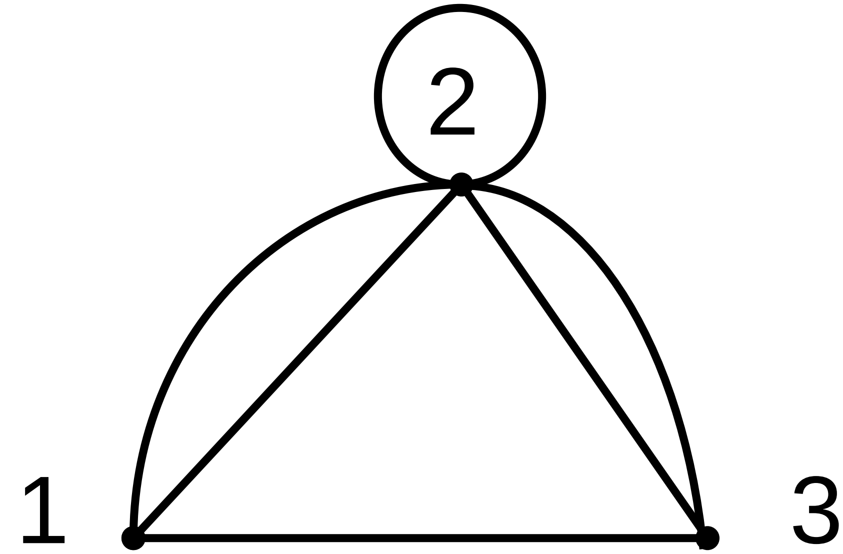
1) Матрица смежности.

Матрица смежности A(G) графа (орграфа)

$$a_{ij} = \begin{cases} 1, & \text{если } v_i \text{ и } v_j - \text{смежные} \\ 0, & \text{иначе} \end{cases}$$

Память $M(n,m)=O(n^2)$.

Пример. Псевдограф



$$A(P) = \begin{pmatrix} 0 & 2 & 1 \\ 2 & 2 & 2 \\ 1 & 2 & 0 \end{pmatrix}$$

2) <u>Матрица инцидентности.</u>

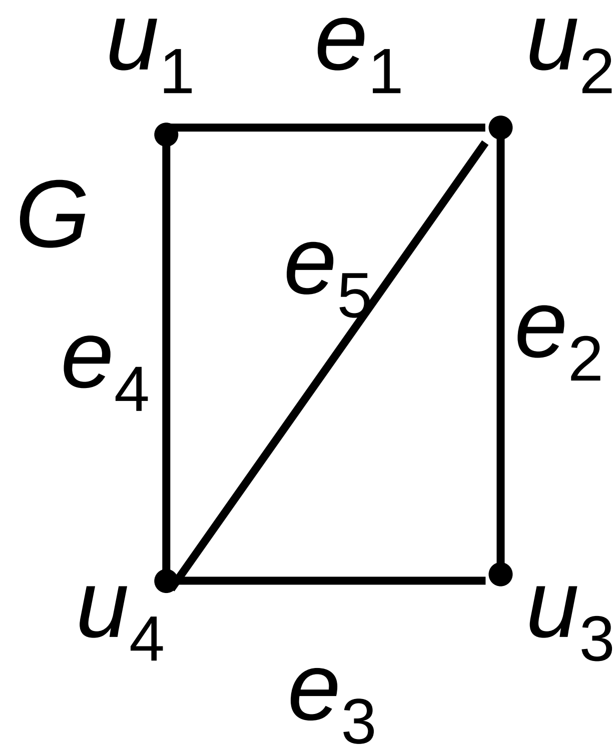
I(G), имеет *n* строк и *m* столбцов,

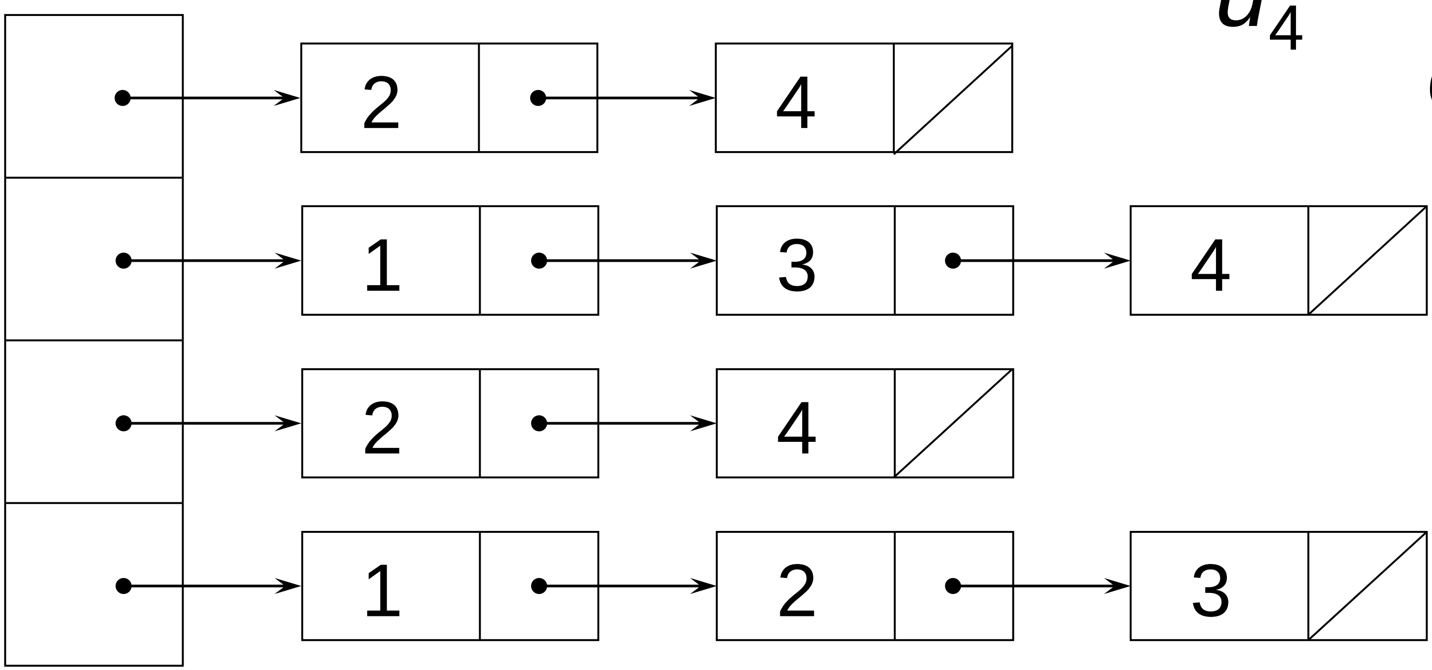
$$i_{kl} = \begin{cases} 1, & \text{если вершина } v_k \text{ инцидентна ребру } e_l \\ 0, & \text{иначе} \end{cases}$$

$$i_{kl} = \begin{cases} 1, \text{ если } v_k \text{ конец ребра } e_l \\ 0, \text{ если } v_k \text{ и ребро } e_l \text{ не инцидентны} \\ -1, \text{ если } v_k \text{ начало ребра } e_l \end{cases}$$

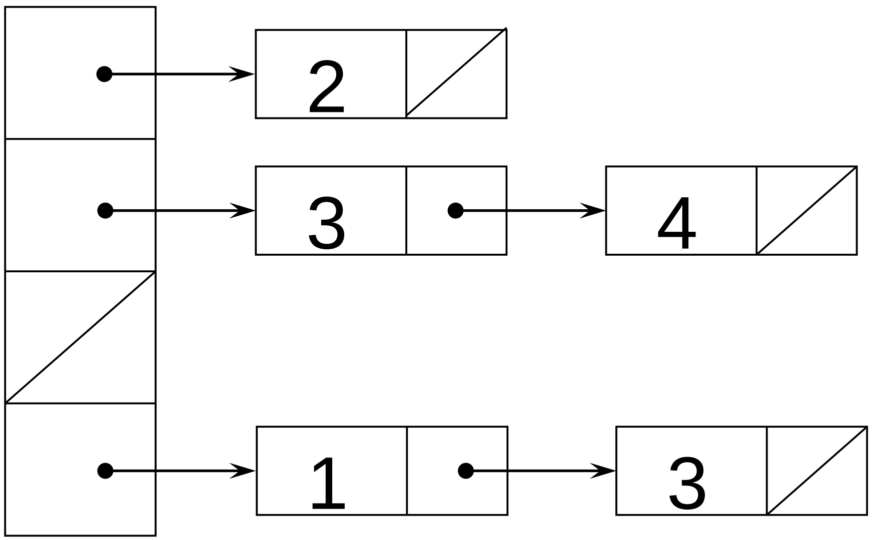
$M(n,m)=O(n \cdot m)$.

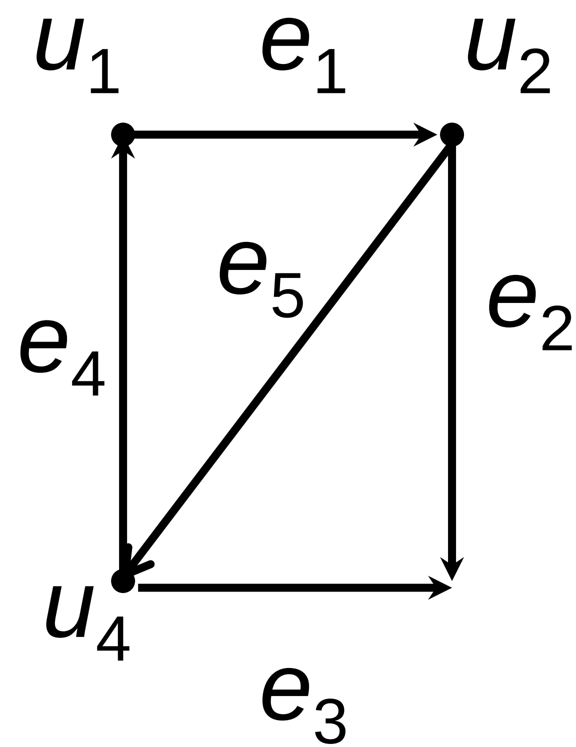
3) Списки смежности.





Для орграфа





Для неориентированных графов

$M(n,m)=O(n+2m)$,

для орграфов $M(n,m)=O(n+m)$.

4) <u>Массив ребер (дуг).</u>

Граф G		Граф D	
начало	конец	начало	конец
1	2	1	2
1	4	2	3
2	3	2	4
2	4	4	1
3	4	4	3

3.11 Обходы графов

Обход графа – это некоторое систематическое перечисление его вершин (ребер).

Алгоритм:

1. Сначала все вершины считаются неотмеченными.
2. Выбирается любая вершина (начало поиска), заносится в структуру данных T и помечается.
3. Следующие действия выполняются в цикле до тех пор, пока структура T не станет пустой:

   - из структуры данных T выбирается вершина *u (*и удаляется*)*;

   - она выдается в качестве очередной пройденной вершины;

   - перебираются все вершины из Γ(u), и все те, которые не помечены, тоже заносятся в структуру T и помечаются.

Если T – это стек (LIFO), то обход называется поиском в глубину

Если T – это очередь (FIFO), то обход называется поиском в ширину.

Любой из рассмотренных обходов позволяет построить остовное дерево исходного графа с корнем в исходной вершине

Теорема Если граф G связен и конечен, то поиск в ширину или поиск в глубину обойдет все вершины графа по одному разу.

Доказать надо три факта:

1) Единственность обхода
2) Завершаемость алгоритма
3) Обход все вершин

3.12 Выделение компонент связности в орграфах

Компоненты связности графа *G* – это его максимальные связные подграфы

Для выделения компонент связности графа можно в качестве основы алгоритма использовать метод поиска в глубину.

Компоненты сильной связности (КСС) орграфа G – это его максимальные сильно связные подграфы.

Если вершина не связана с другими, то считается, что она сама образует КСС.

Орграф, который получается стягиванием в одну вершину каждой КСС, называется фактор-графом, или конденсацией орграфа G.

Для выделения КСС орграфа можно в качестве основы алгоритма использовать метод поиска в глубину.

В качестве примера рассмотрим Алгоритм Косараджу–Шарира (Kosaraju, Sharir)

1 этап: запускается поиск в глубину на обращении графа

2 этап: запускается поиск в глубину на исходном графе в порядке, определяемом списком, полученным на первом этапе (в обратном порядке)