

ОБХОДЫ

Основные разделы:

- Поиск в глубину;
- Поиск в ширину.

Обход графа — некоторое систематическое перечисление его вершин или ребер.

Среди вершинных обходов наиболее известны «поиск в глубину» и «поиск в ширину».

На их основе сформулированы многие из встречающихся далее алгоритмов.

К реберным обходам относится эйлеров цикл.

Поиск в глубину

Идея алгоритма.

Выходя из начальной вершины, строим простую цепь. Пройденным вершинам приписываем метки. Продолжаем строить цепь до тех пор, пока не встретим уже отмеченную вершину или не попадем в висячую вершину. В этом случае возвращаемся на шаг назад и выбираем ребро, ведущее к непомеченной вершине. Процесс продолжаем до тех пор, пока всем вершинам не будут присвоены метки.

Пусть $G(V, E)$ — связный неориентированный граф. Для каждой вершины будем запоминать ее метку $s(v)$ и вершину, из которой попали в данную (предшественника) $p(v)$.

Кроме того, в описании алгоритма будут использоваться следующие обозначения: v^* — текущая вершина, V^* — множество еще не помеченных вершин, $\Gamma(v^*)$ — окрестность вершины v^* .

Шаг 0. $u \in V$ — произвольная вершина графа.

Приписываем ей метку 1.

$$s(u) := 1, \quad s := 1, \quad v^* := u, \quad V^* := V \setminus \{u\}$$

Шаг 1. Алгоритм продолжает работу до тех пор,
пока $|V^*| \neq 0$. Если $V^* \cap \Gamma(v^*) = \emptyset$, то переходим на шаг 2.

Если $V^* \cap \Gamma(v^*) \neq \emptyset$, то мы выбираем $v \in V^* \cap \Gamma(v^*)$ и
приписываем ей очередную метку:

$$s(v) := s + 1, \quad s := s + 1, \quad p(v) := v^*, \quad V^* := V^* \setminus \{v\},$$

$$v^* := v.$$

Переходим на начало шага 1.

Шаг 2. $V^* \cap \Gamma(v^*) = \emptyset$, и мы «возвращаемся назад», т.е.

$$v^* := p(v).$$

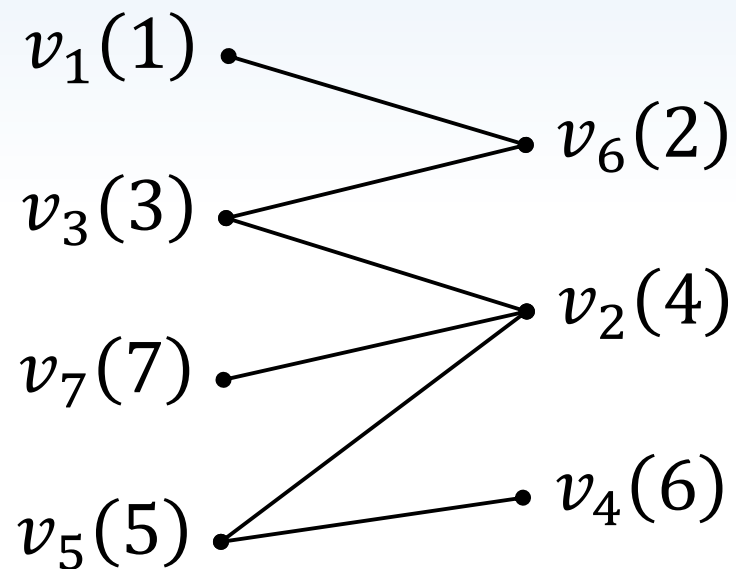
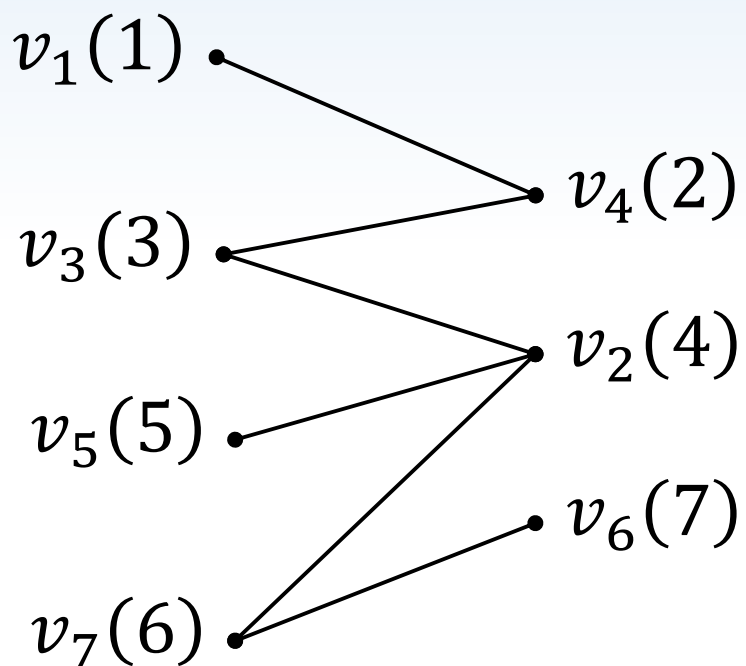
Переходим на шаг 1.

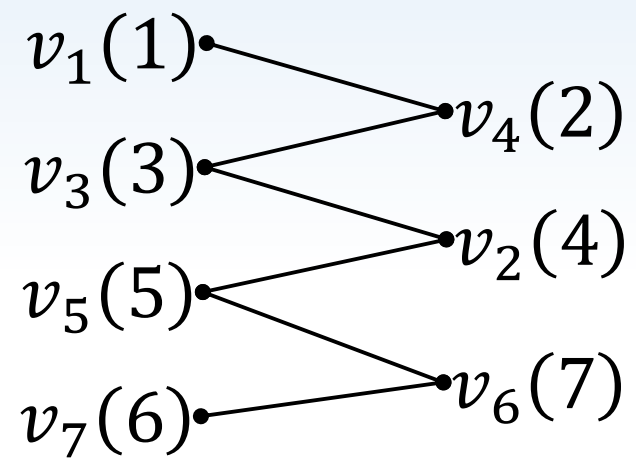
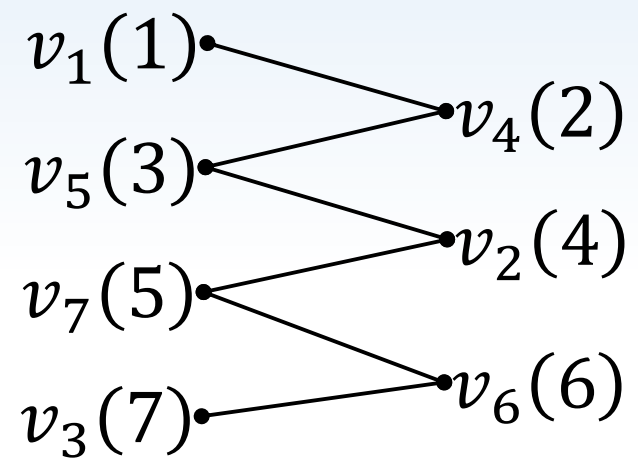
- **Замечание 1.** Если граф несвязный, то поиск в глубину осуществляется для вершин, достижимых из начальной.
- **Замечание 2.** Поиск может быть осуществлен и на орграфе. В этом случае необходимо учесть направление дуг.

ПРИМЕР: В заданном матрицей смежности графе провести поиск в глубину из вершины v_1 :

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

Решение: Поиск может быть осуществлен несколькими способами, некоторые из них приведены на рисунках. В скобках указана метка вершины.

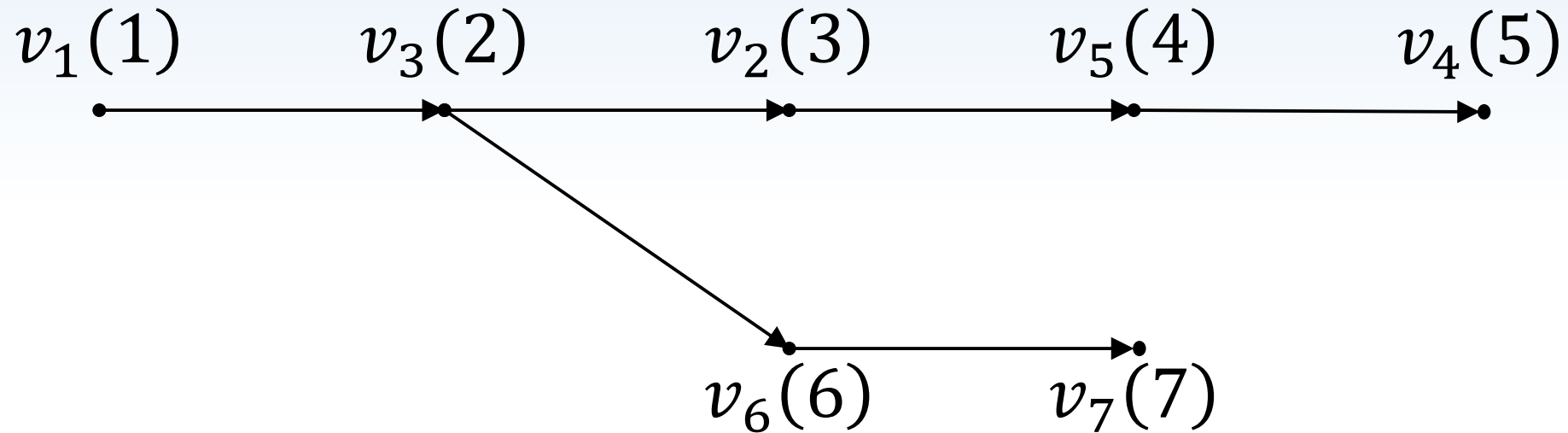




ПРИМЕР: В заданной матрице смежности орграфе обойти все вершины, используя поиск в глубину.

$$A(\vec{G}) = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Решение: Проводим поиск из первой вершины. В скобках указана метка вершины.



Поиск в ширину

Идея алгоритма.

Выходя из начальной вершины, помечаем все вершины из ее окрестности, как вершины первого «уровня». Затем, у каждой из вершин первого уровня помечаем еще не отмеченные вершины ее окрестности, как вершины второго уровня и т.д.. У вершин уровня k помечаем еще не отмеченные вершины окрестности, как вершины уровня $k + 1$. Процесс продолжаем до тех пор, пока все вершины не получат метки.

Пусть $G(V, E)$ — связный неориентированный граф. Для каждой вершины будем запоминать ее метку (уровень) $s(v)$. Кроме того, V^* — множество еще не помеченных вершин, $\Gamma(v)$ — окрестность вершины v , C_i — множество вершин с метками $s(v) = i$. В итоге множество вершин V будет разбито на классы (уровни) C_i .

Шаг 0. $v \in V$ — начальная вершина поиска. Приписываем ей метку 0.

$$s(v) := 0, \quad C_0 = \{v\}, \quad V^* := V \setminus C_0.$$

Шаг 1. Всем вершинам из окрестности v присваиваем метку 1 и объединяем их в класс C_1 :

$$\forall v \in V^* \cap \Gamma(v), s(v) := 1, C_1 = \{v | s(v) = 1\},$$

$$V^* := V^* \setminus C_1.$$

Шаг k+1. Пусть $\Gamma(C_k) = \cup \Gamma(v)$, $v \in C_k$.

Всем вершинам из множества $V^* \cap \Gamma(C_k)$ присваиваем метку $k + 1$ и объединяем их в класс C_{k+1} :

$$\forall v \in V^* \cap \Gamma(C_k), \quad s(v) := k + 1,$$

$$C_{k+1} = \{v | s(v) = k + 1\}, \quad V^* := V^* \setminus C_{k+1}$$

Процесс продолжаем до тех пор, пока V^* не пусто.

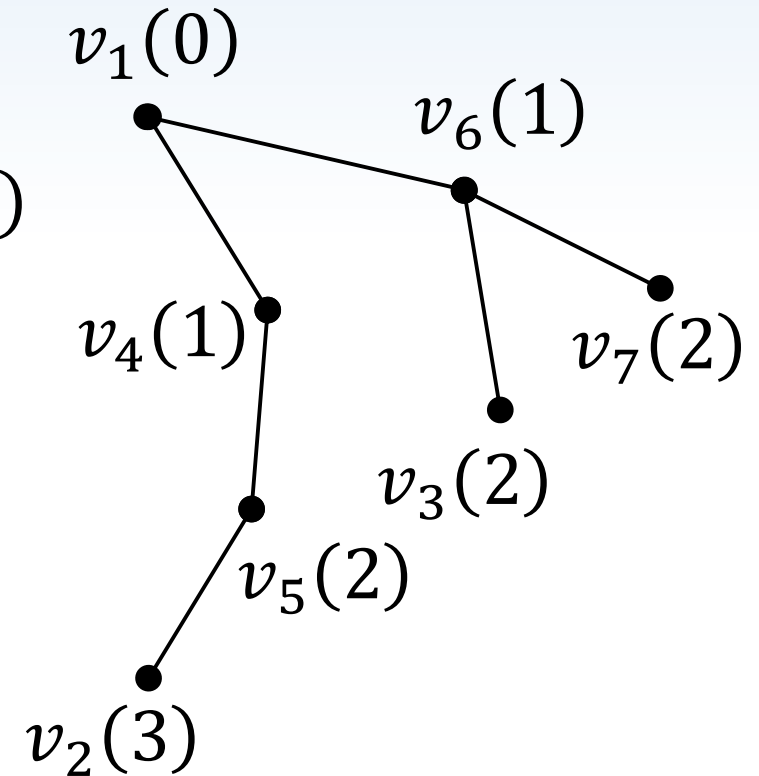
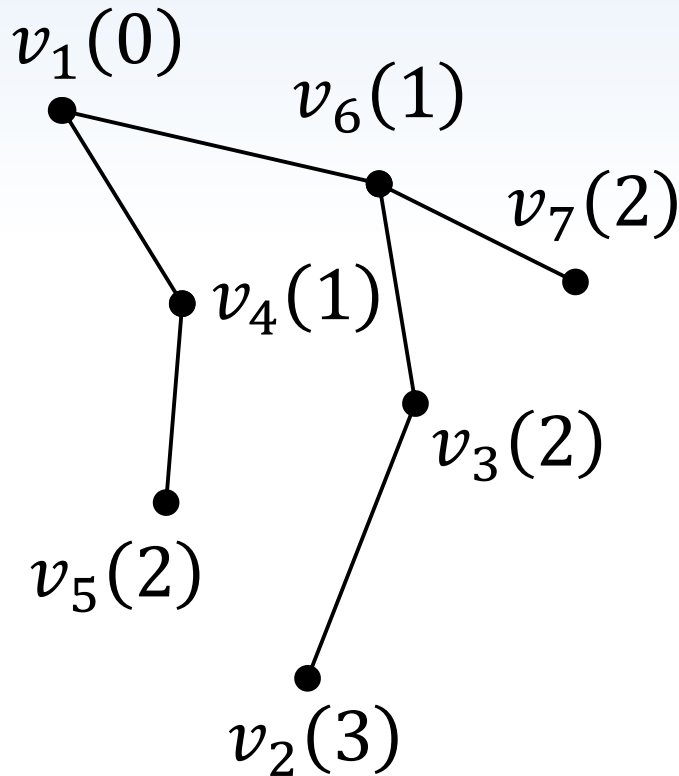
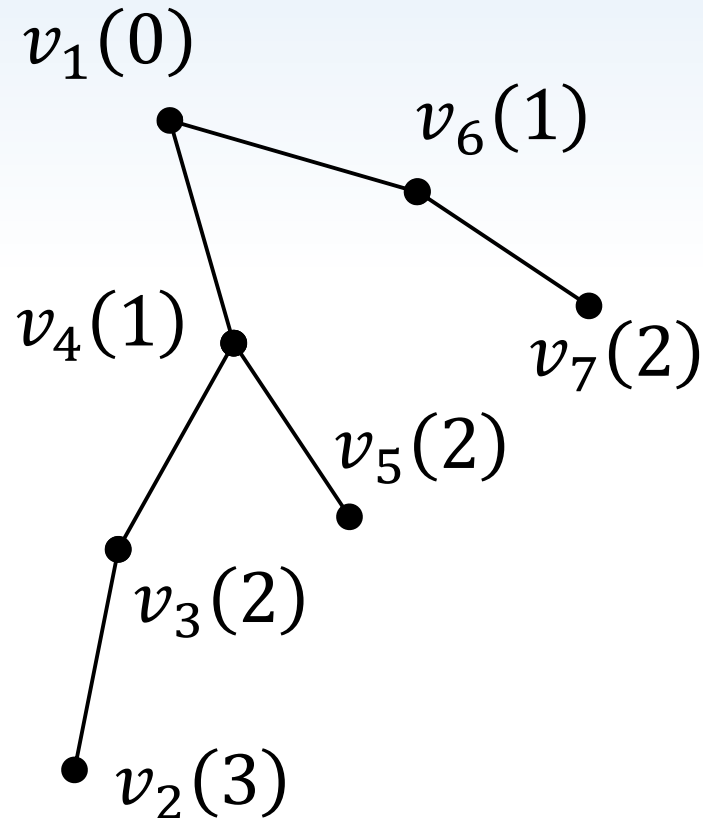
- **Замечание 1.** Если граф несвязный, то поиск в ширину осуществляется для вершин, достижимых из начальной.
- **Замечание 2.** Поиск в ширину можно провести из множества вершин $X \subset V(G)$. В этом случае, каждый шаг алгоритма выполняется для всех вершин заданного множества по очереди: вершинам данного множества X присваиваем метку 0, вершинам из окрестности множества X — метку 1 и т.д. Процесс продолжаем до тех пор, пока все вершины, достижимые из вершин множества X , не получают метки.

- **Замечание 3.** Поиск может быть осуществлен и на орграфе. В этом случае необходимо учесть направление дуг.

ПРИМЕР: В заданной матрице смежности графа провести поиск в ширину из первой вершины:

$$A(\vec{G}) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Решение: Поиск может быть осуществлен несколькими способами, некоторые из них приведены на рисунке. В скобках указан уровень (метка) вершины.



Как видно, уровень вершины не зависит от способа поиска. Фактически, метка вершины равна расстоянию от нее до вершины, из которой идет поиск (в данном случае — до первой).

Расстояние между вершинами графа — это количество ребер кратчайшей длины, соединяющих вершины.

ПРИМЕР: В заданном матрицей смежности $A(\vec{G})$ орграфе провести:

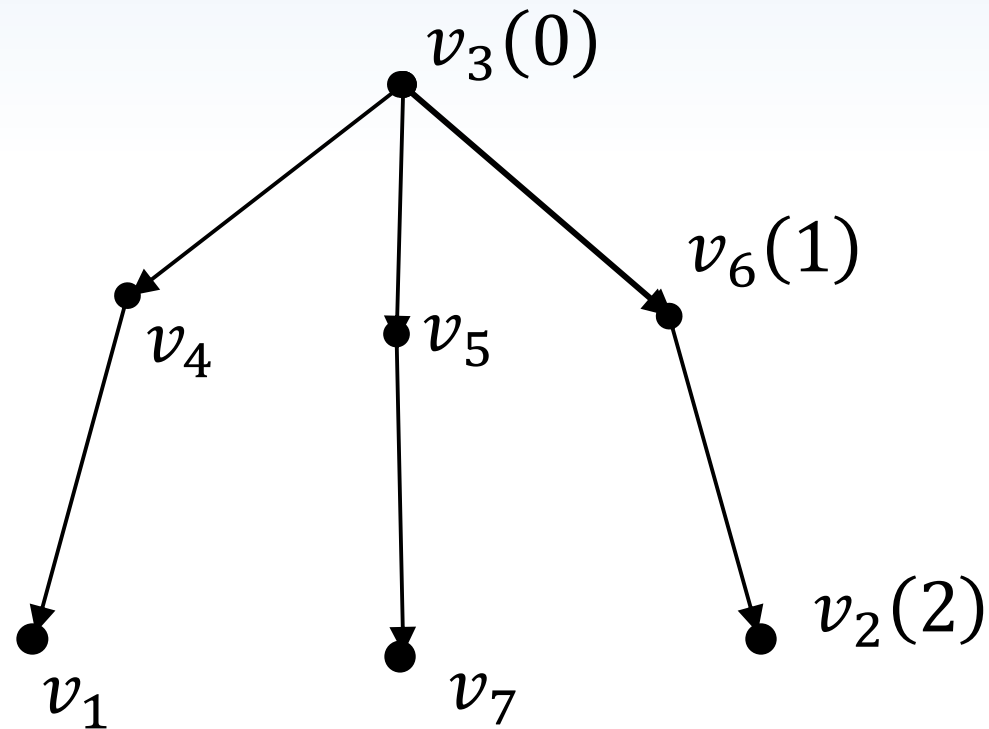
- 1) поиск в ширину из вершины v_3 ;
- 2) поиск в глубину из вершины v_3 ;
- 3) поиск в ширину из множества вершин $\{v_1, v_3\}$.

$$A(\vec{G}) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Решение:

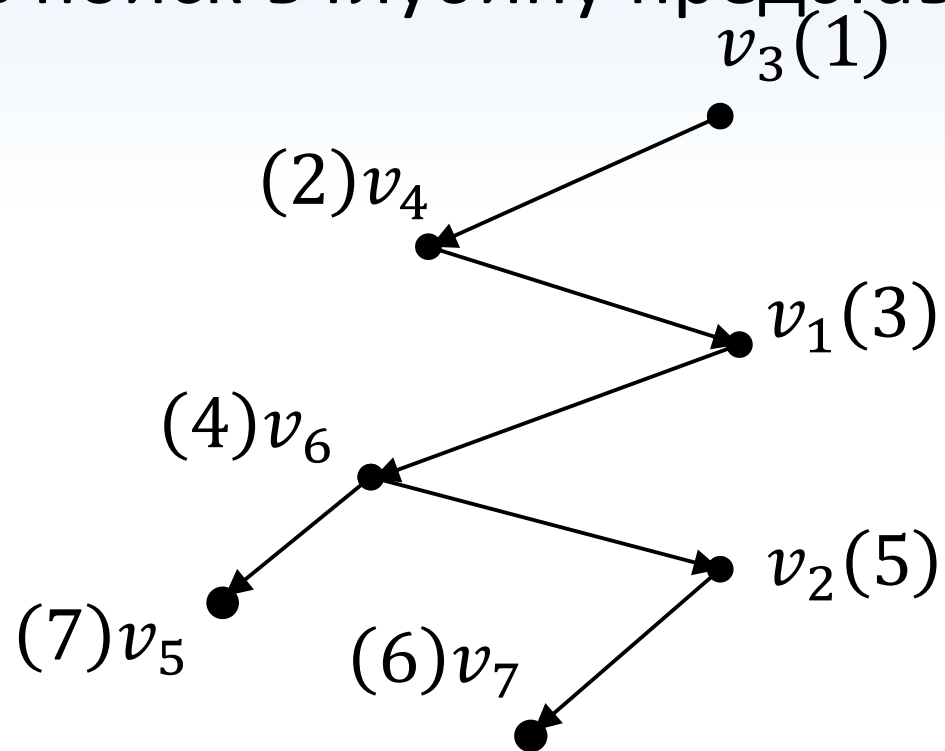
1) Из вершины v_3 ведут дуги в v_4 , v_5 и v_6 (им соответствуют единицы в третьей строке матрицы смежности). Приписываем вершинам v_4 , v_5 , и v_6 метку 1 и рассматриваем окрестность каждой из них. Из v_4 ведут дуги в v_1 , v_3 и v_5 . Но вершины v_3 и v_5 уже рассмотрены ранее, поэтому остается только вершина v_1 , которой присваивается метка 2. Из v_5 ведут ребра в v_1 , v_6 и v_7 . Вершины v_1 и v_6 рассмотрены ранее, присваиваем метку 2 вершине v_7 . Из v_6 ведут дуги в v_2 , v_3 , v_5 , и v_7 .

Ранее не рассматривалась только v_2 , присваиваем ей метку 2. Заметим, что при рассмотрении вершин из окрестности вершины v_3 в другом порядке (например, v_6, v_5, v_4) метки приписанные им были бы такими же, а вот дерево, иллюстрирующее поиск могло бы выглядеть иначе.



2) Из вершины v_3 переходим в вершину с наименьшим номером — v_4 (присваиваем ей метку 2). Из v_4 ведут дуги в v_1 , v_3 и v_5 . Наименьший номер у v_1 (присваиваем ей метку 3). Из v_1 ведут дуги в v_4 , и v_6 , но v_4 рассмотрена ранее, поэтому метку 4 присваиваем вершине v_6 . Из v_6 ведут дуги в v_2 , v_3 , v_5 и v_7 . Вершина v_2 не рассматривалась ранее, присваиваем ей метку 5. Из v_2 ведут дуги в v_1 , и v_7 . Присваиваем вершине v_7 метку 6. Из v_7 ведут дуги в v_2 , v_3 , и v_6 . Все эти вершины были рассмотрены ранее, поэтому “поднимаемся” в вершину v_2 .

В вершине v_2 ситуация та же: все вершины из ее окрестности помечены ранее, “поднимаемся” - в вершину v_6 . Из непомеченных вершин смежных с v_6 наименьший номер у v_5 . Присваиваем ей метку 7. Все вершины графа помечены. Дерево, иллюстрирующее поиск в глубину представлено на рисунке.



3) Вершинам v_1 и v_3 приписываем метку 0. Из вершины v_1 ведут дуги в v_4 и v_6 , из вершины v_3 ведут дуги в v_4 , v_6 и v_5 . Приписываем вершинам v_4 , v_5 и v_6 метку 1 и рассматриваем окрестность каждой из них. Из вершин v_4 , v_5 и v_6 ведут дуги во все вершины орграфа, кроме v_4 . На данном шаге осталось две не рассмотренных ранее вершины v_2 и v_7 , которым присваивается метка 2. Все вершины графа помечены (рисунок).

