**Лабораторная работа 5. АЛГОРИТМЫ НА ГРАФАХ**

**(алгоритмы поиска в ширину и глубину, топологическая сортировка)**

**ЦЕЛЬ РАБОТЫ:** Освоить сущность и программную реализацию: а) способов представления графов; б) алгоритмов поиска в ширину и глубину; в) алгоритма топологической сортировки графов.

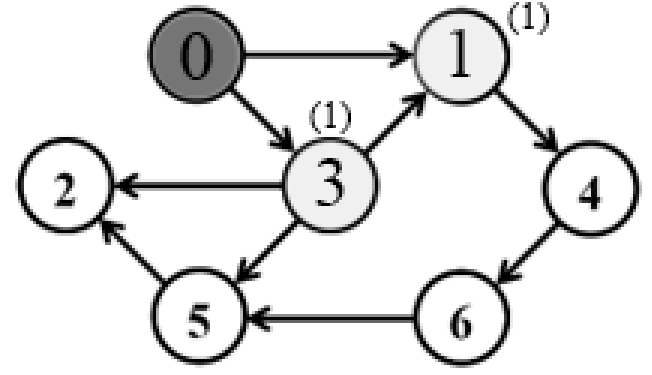
**ЗАДАНИЕ ДЛЯ ВЫПОЛНЕНИЯ:**

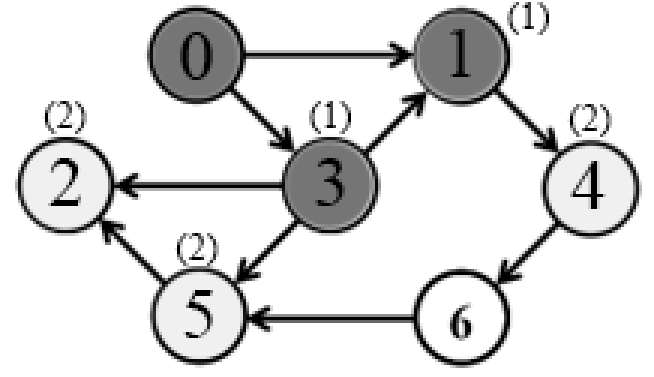
**Задание 1.**  Ориентированный граф **G** взять в соответствии с вариантом. Осуществить алгоритмы поиска в ширину и глубину, а также алгоритма топологической сортировки аналогично примерам, рассмотренным на лекциях. Оформить отчет, включив в него **каждый** шаг выполнения алгоритмов.

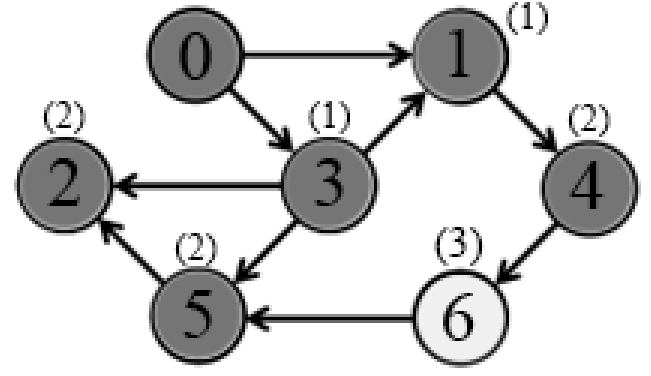
|  |  |
| --- | --- |
| Вариант | Граф G |
| 1 |  |

**В ширину (Breadth-first search)**

В этом способе при выборе очередной вершины предпочтение отдается ближайшей.

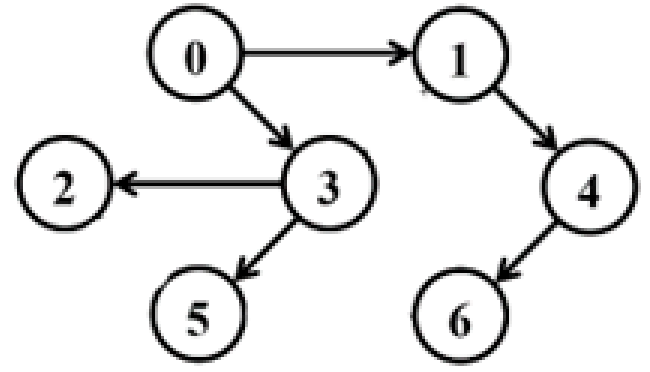
Как начальную вершину возьмем 0. Вершины из окрестности 0 помечаем как вершины 1 уровня.

Отмечаем все окрестности вершин 1 уровня.

Отмечаем все окрестности вершин 2 уровня.

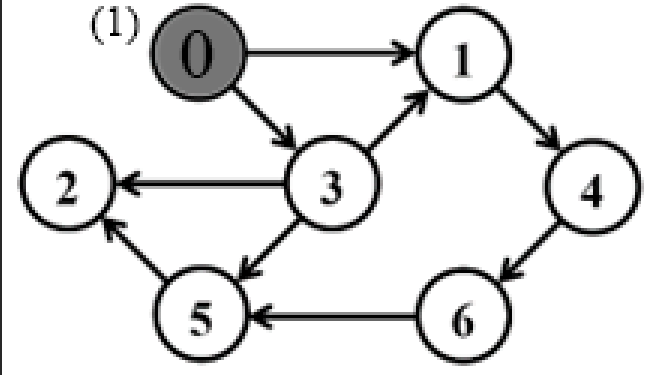
Все вершины графа помечены. Обход завершен.

Получаем последовательность 0-1-3-4-2-5-6.



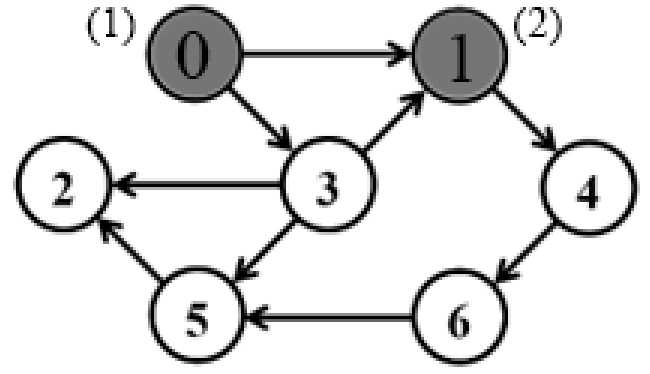
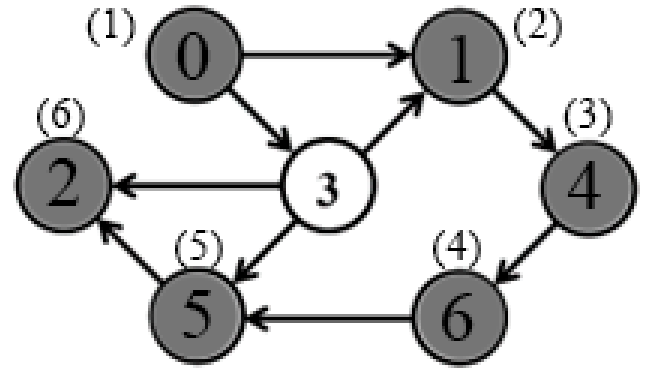
**В глубину (Depth-first search)**

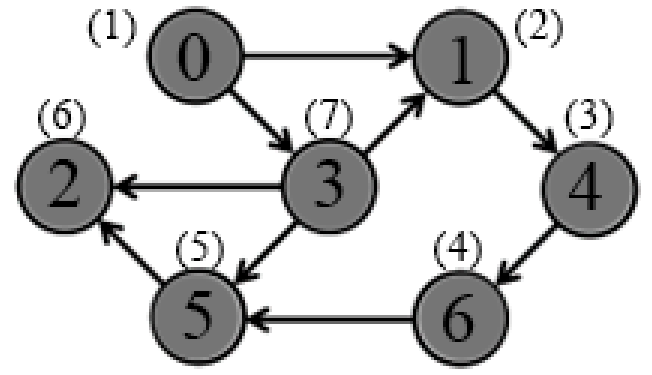
В этом способе для каждой непройденой вершины, необходимо найти все смежные вершины и повторить поиск для каждой.

В качестве стартовой вершины выберем нулевую вершину для нее устанавливаем значение метки 1.

Переходим к следующей вершине и присваиваем ей метку 2. Последовательность 1.

Доводим последовательность до конца.



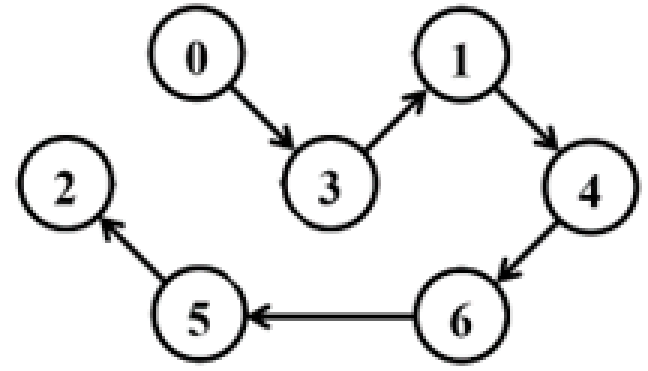


Получаем последовательность 2-5-6-4-1.

Переходим к другой вершине.

Получаем последовательность 3-0.

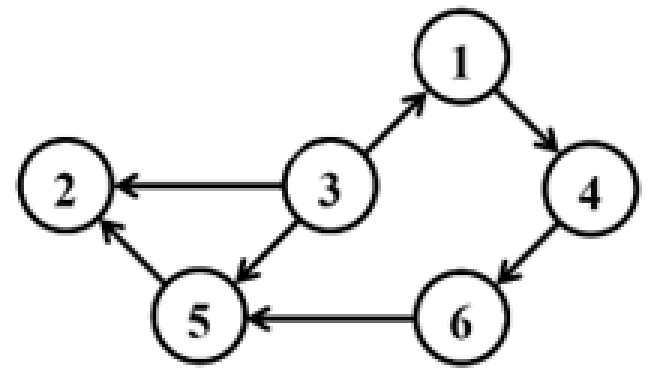
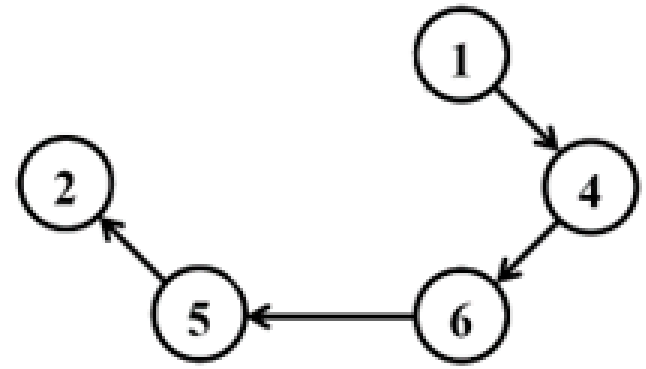
Из двух последовательностей получаем: 2-5-6-4-1-3-0.

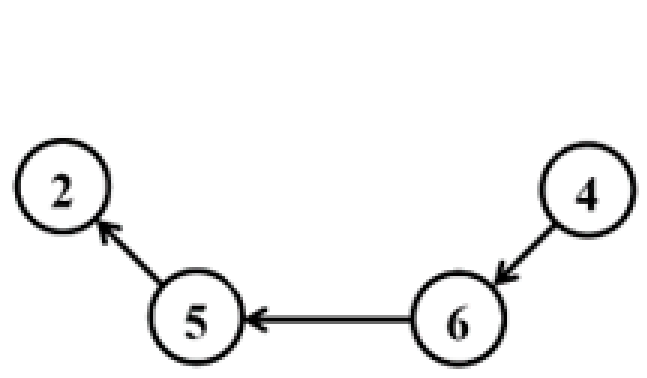


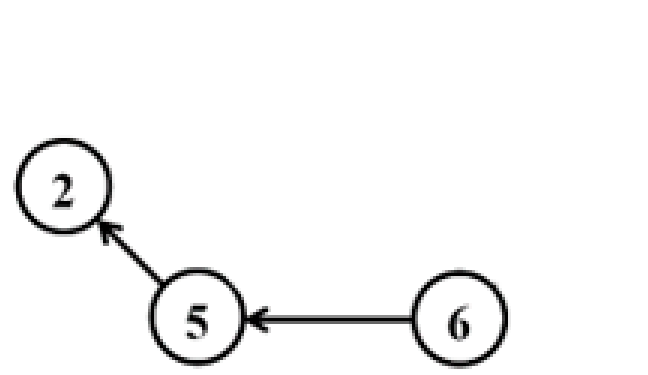
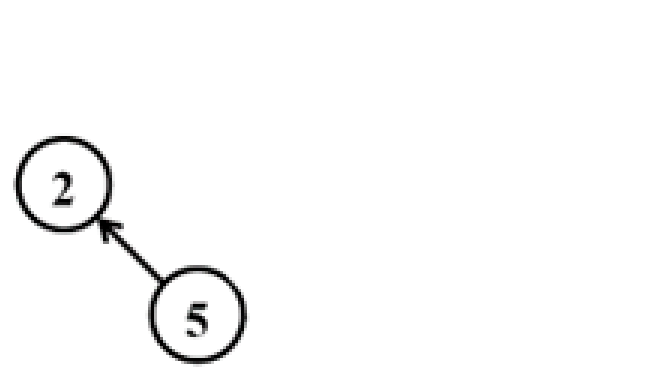
**Топологическая сортировка**

Топологическая сортировка будет выполнена с помощью метода Демукрона.

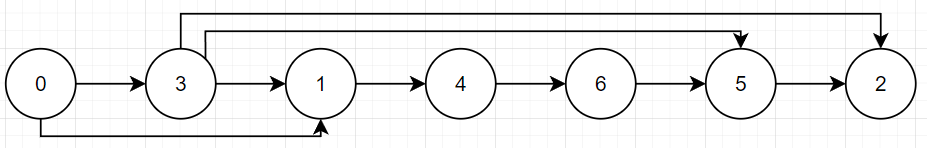
В исходном графе вершины, являющиеся истоками, будут вычеркиваться, а также дуги, связанные с ними. Каждому такому удалению будет соответствовать запись вычеркнутых вершин на новый уровень обновленного графа.

В исходном графе исток – нулевая вершина. Ее и вычеркиваем. Записываем эту вершину на нулевой уровень отсортированного графа.



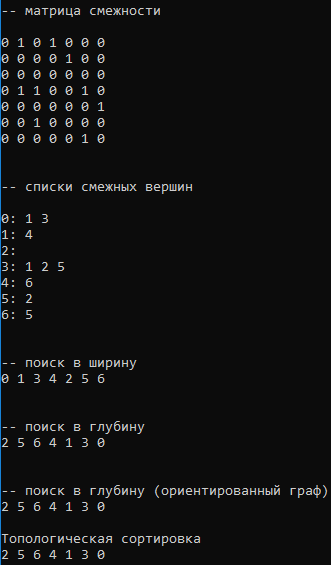




После исключения всех истоков получим отсортированный граф. В этом графе все вершины будут соединены точно так же, как и в исходном.  **Задание 2.** Осуществить программную реализацию алгоритмов на C++. Разработать структуры **AMatrix** и **АList** для представления ориентированного графа матричным и списковым способом. Разработать функции преобразования из одного способа представления в другой. Разработать функцию **BFS** обхода вершин графа, используя метод поиска в ширину. Продемонстрировать работу функции. Копии экрана вставить в отчет.

**Задание 3.** Разработать функцию **DFS** обхода вершин графа, используя метод поиска глубину. Продемонстрировать работу функции. Копии экрана вставить в отчет.

**Задание 4.**Доработайте функцию **DFS**,для выполнения топологической сортировки графа. Продемонстрировать работу функции. Копии экрана вставить в отчет.



**Вывод:** освоил сущность и программную реализацию: а) способов представления графов; б) алгоритмов поиска в ширину и глубину; в) алгоритма топологической сортировки графов.