Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра информационных систем и технологий

Отчёт по лабораторной работе №5

«Алгоритмы на графах»

по дисциплине

«Математическое программирование»

Выполнил:

студент 2-го курса спец. ДЭиВИ

Глушкова М.Е.

Вариант №2

Проверил:

ассистент кафедры

Барковский Е.В.

**Лабораторная работа 5. АЛГОРИТМЫ НА ГРАФАХ**

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

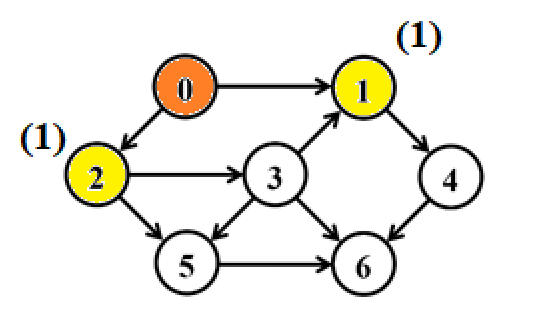
**Цель работы:** Освоить сущность и программную реализацию: а) способов представления графов; б) алгоритмов поиска в ширину и глубину; в) алгоритма топологической сортировки графов.

**Задание для выполнения:**

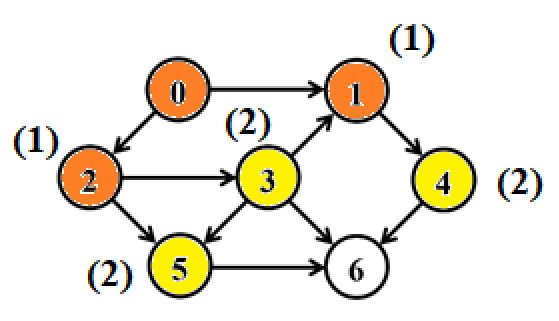
**Задание 1.**  Ориентированный граф **G** взять в соответствии с вариантом. Осуществить алгоритмы поиска в ширину и глубину, а также алгоритма топологической сортировки аналогично примерам, рассмотренным на лекциях. Оформить отчет, включив в него **каждый** шаг выполнения алгоритмов.

|  |  |
| --- | --- |
| Вариант | Граф G |
| 2 |  |

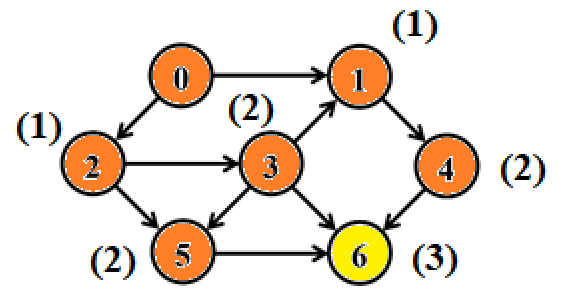
В качестве начальной вершины выберем нулевую. Помечаем все вершины из ее окрестности, как вершины первого «уровня».



Затем помечаем все неотмеченные вершины из окрестности вершин с меткой 1.



Помечаем оставшуюся неотмеченную вершину из окрестности вершин с меткой 2.

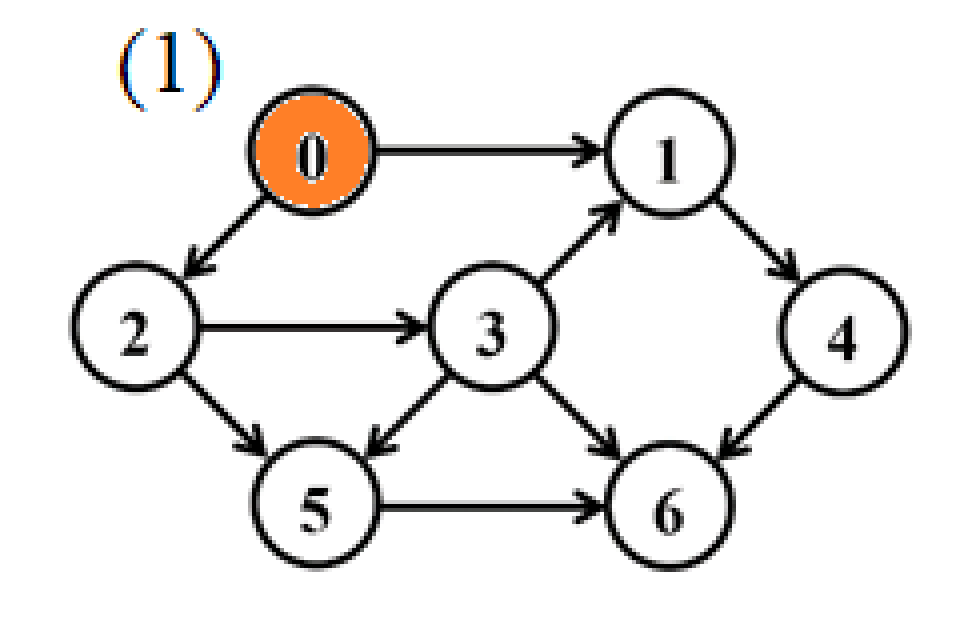


Таким образом все вершины оказались помеченными. Обход завершен.

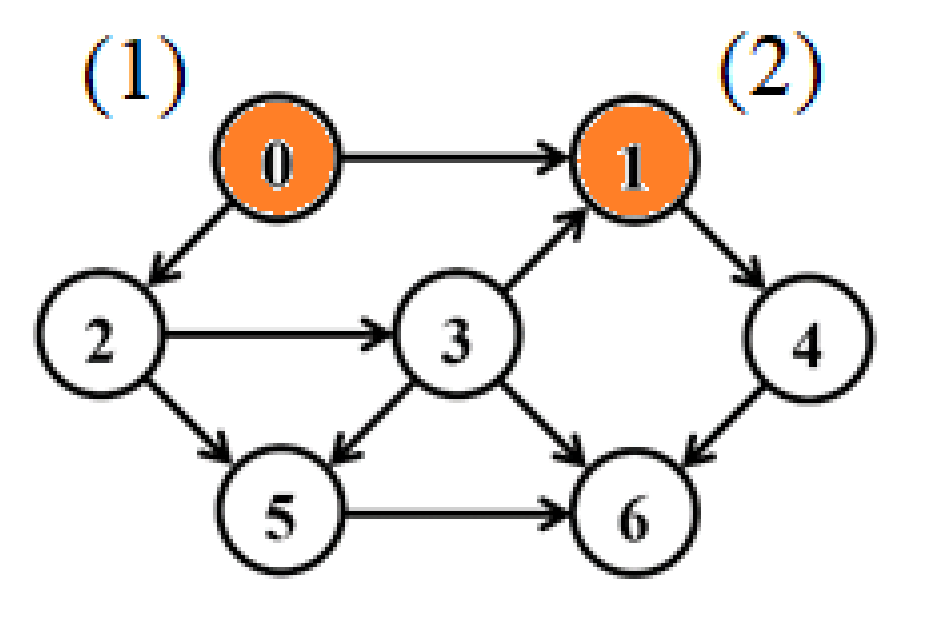
В итоге получаем последовательность 0 – 1 – 2 – 4 – 3 – 5 – 6.

**Обход в глубину**

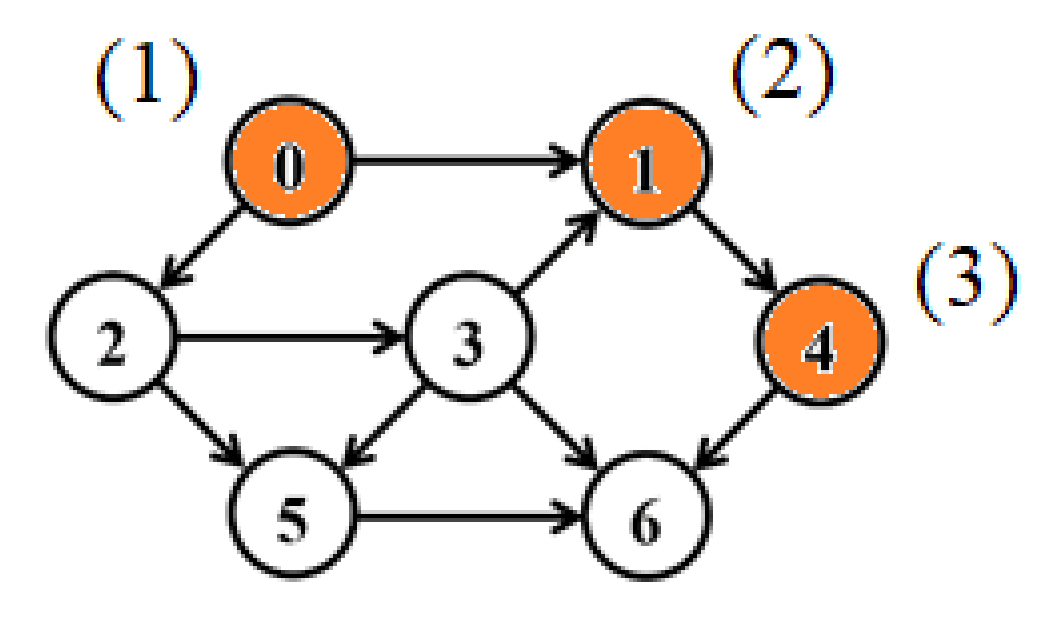
В качестве стартовой вершины выберем нулевую вершину для нее устанавливаем значение метки 1.

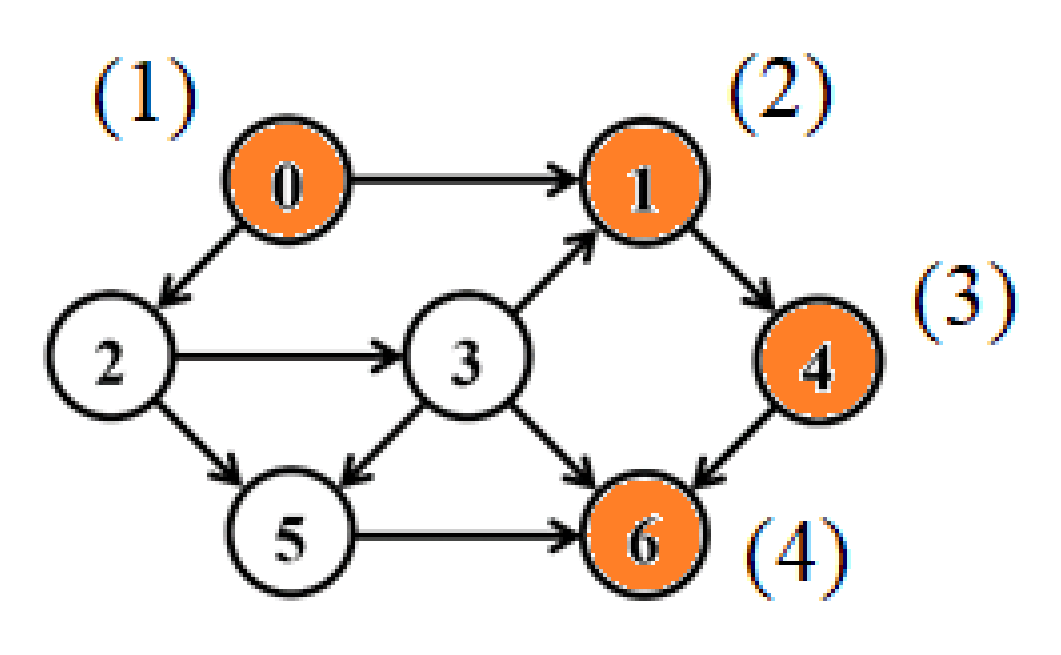


Переходим к следующей вершине и присваиваем ей метку 2. Последовательность 1.

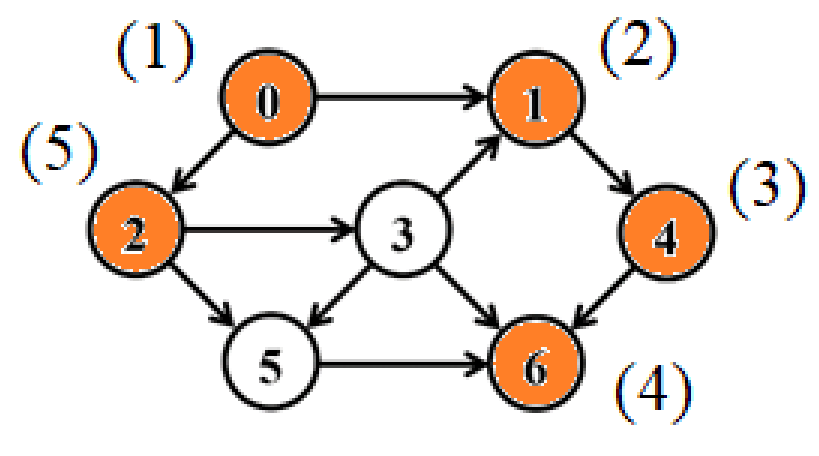


Далее продолжаем действия.

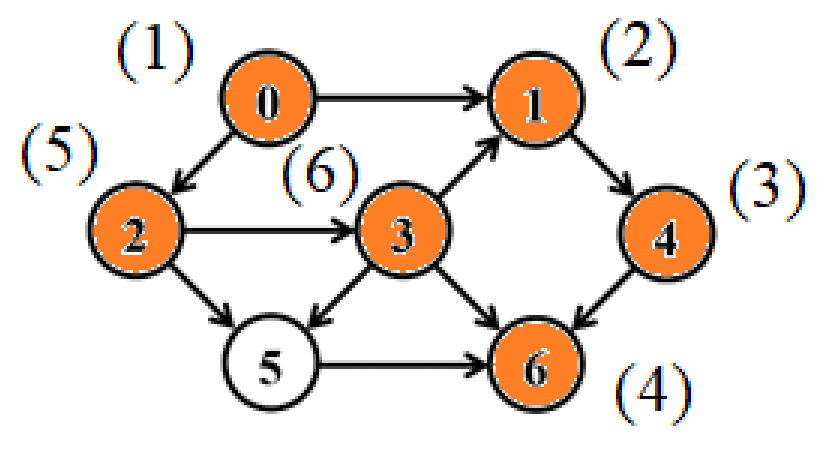


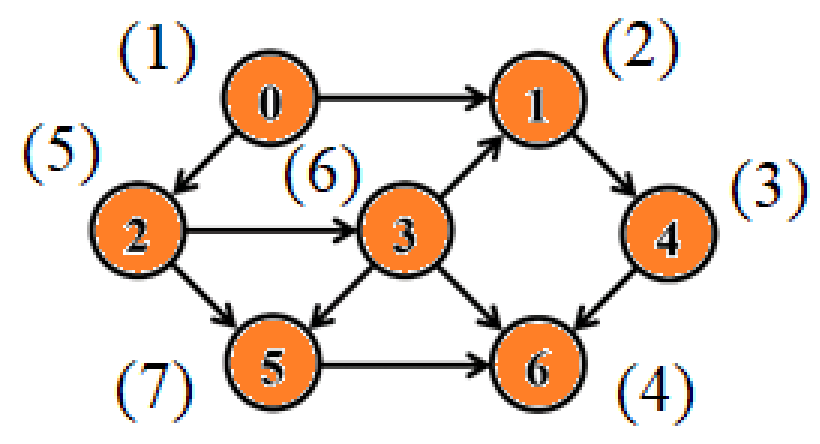


Таким образом получается последовательность 6 – 4 – 1.



Переходим к вершине 2.



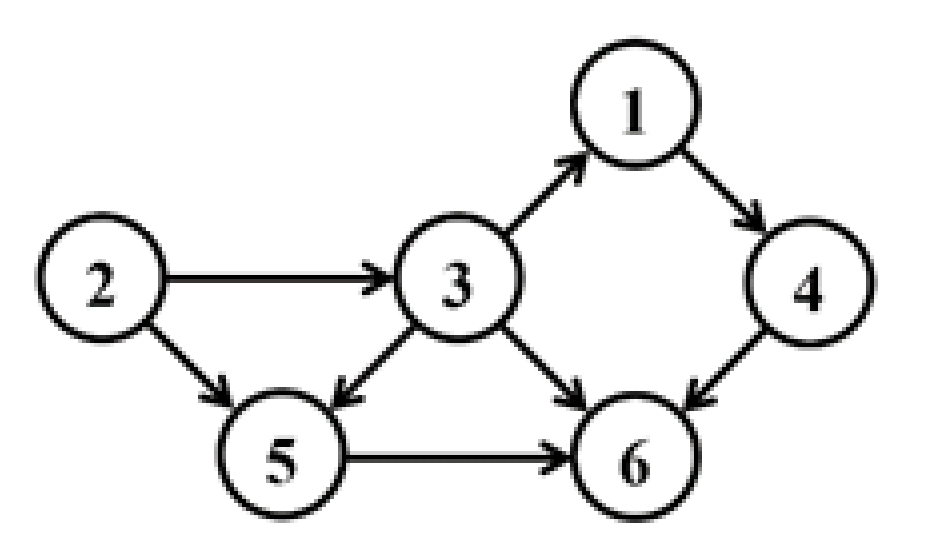


В конце добавляем вершины 2 – 0 и получаем следующую последовательность: 6 – 4 – 1 – 5 – 3 – 2 – 0.

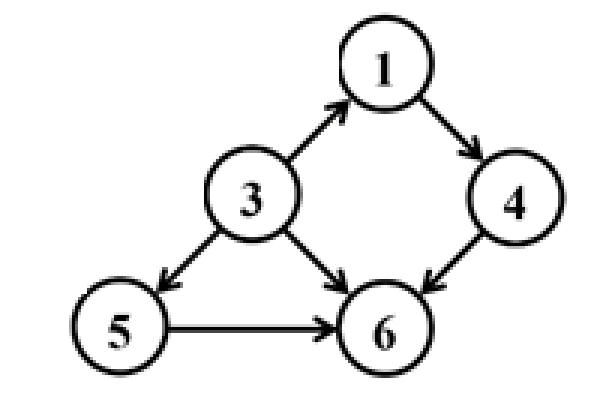
**Топологическая сортировка**

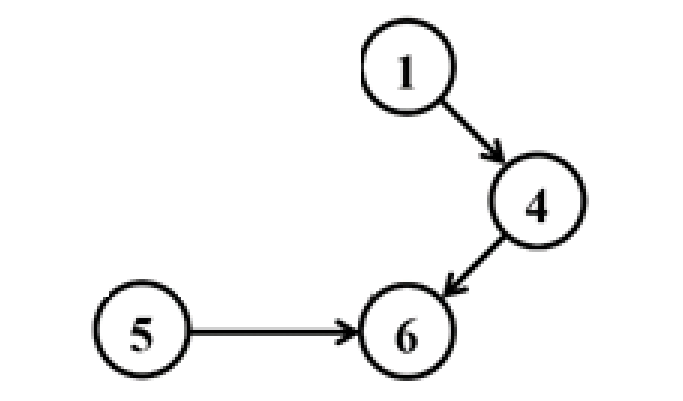
Выполним топологическую сортировку алгоритмом Демукрона. Суть в том, что в исходном графе мы будем вычеркивать вершины, являющиеся истоками, а также дуги, связанные с ними. Каждому такому удалению будет соответствовать запись вычеркнутых вершин на новый уровень обновленного графа.

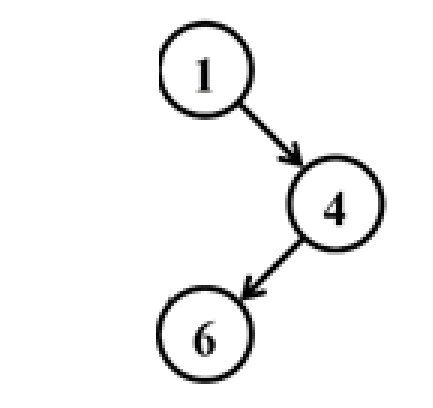
В исходном графе только одна вершина является истоком – нулевая. Запишем ее на нулевой уровень отсортированного графа. Ниже приведен исходный граф с вычеркнутыми вершиной нулевого уровня и связанными с ней дугами, а также выстраивающийся отсортированный граф.

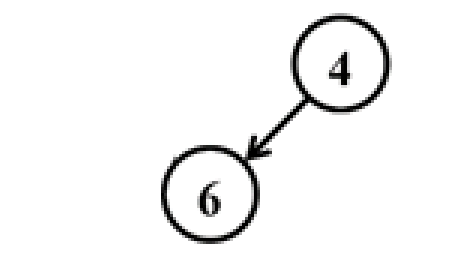


Первоначальным истоком была вершина 0, далее продолжаем повторять действия и убирать истоки, переходя от меньшей вершины к большей.





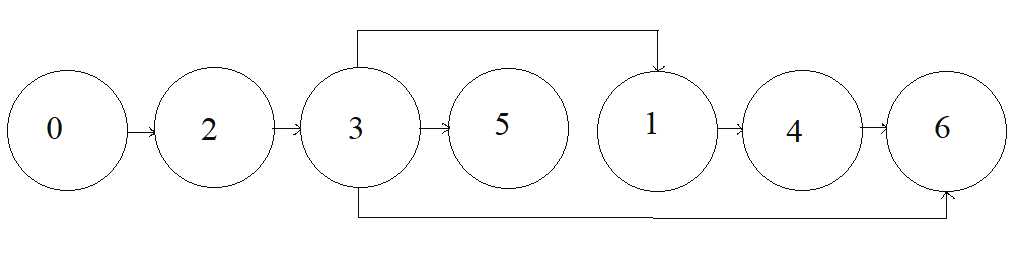






Таким образом получаем следующую топологическую сортировку: 0 – 2 – 3 – 5 – 1 – 4 – 6.

Следующий шаг: соединить вершины в соответствии с исходным графом. Получим следующего вида отсортированный граф.

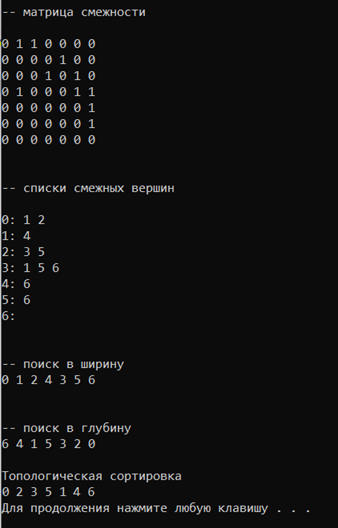


То, что мы прошли все шаги алгоритма до конца, говорит нам о том, что перед нами не циклический граф.

**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:** в ходе данной лабораторной работы освоены сущность и программная реализация: а) способов представления графов; б) алгоритмов поиска в ширину и глубину; в) алгоритма топологической сортировки графов.