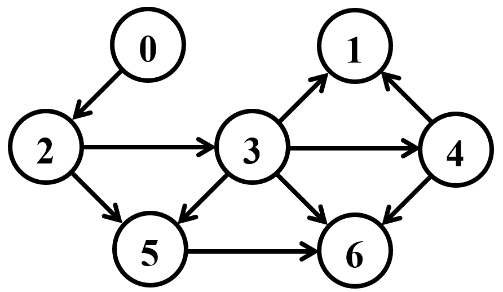
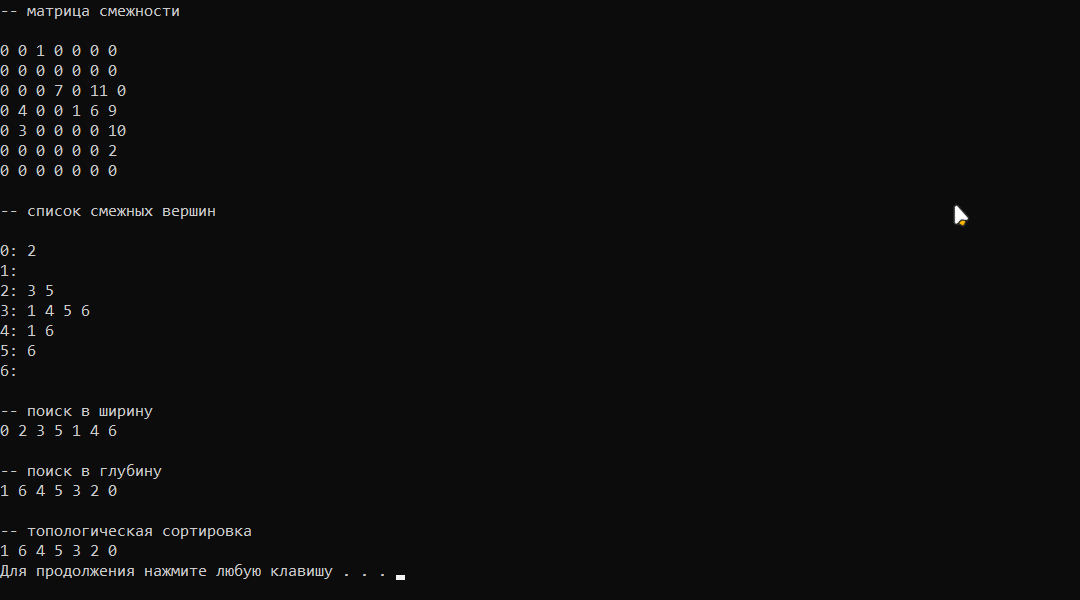
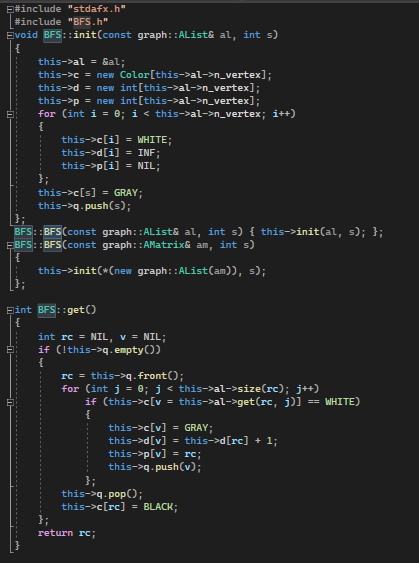
**Лабораторная работа №6. АЛГОРИТМЫ НА ГРАФАХ**

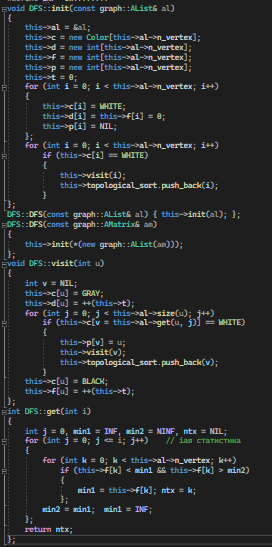
**ЦЕЛЬ РАБОТЫ:** Освоить сущность и программную реализацию: а) способов представления графов; б) алгоритмов поиска в ширину и глубину; в) алгоритма топологической сортировки графов. Разобрать алгоритм Прима и алгоритм Крускала

***Задание 1.*** Ориентированный граф **G** взять в соответствии с вариантом. Представить его в отчете в виде матрицы смежности, матрицы инцидентности, списка смежных вершин.



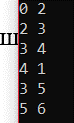






***Задание 6.*** Прима

1. **Инициализация**:
   * Выберем начальную вершину. Для удобства выберем вершину 1.
   * Обозначим вершину 1 как посещенную.
2. **Построение остовного дерева**:
   * Вершина 1: Связана с вершинами 3.
   * Добавляем ребра (1, 3) с весом 1 в список ребер.
3. **Выбор минимального ребра**:
   * Найдем ребро с минимальным весом из списка ребер. Это ребро (1, 3) с весом 1.
4. **Добавление ребра к остову**:
   * Добавим ребро (1, 3) к остову.
5. **Обновление списка ребер**:
   * Удалим ребро (1, 3) из списка ребер.
   * Добавим ребра из вершины 3: (3, 4) с весом 7, (3, 6) с весом 11.
6. **Повторение**:
   * Продолжим поиск минимальных ребер и их добавление к остову.
7. **Шаги алгоритма**:
   * Выбираем ребро (3, 4) с весом 7.
   * Добавляем ребро (3, 4) к остову.
   * Обновляем список ребер: Добавляем ребра из вершины 4: (4, 2) с весом 4, (4, 5) с весом 1, (4, 6) с весом 9.
   * Выбираем ребро (4, 5) с весом 1.
   * Добавляем ребро (4, 5) к остову.
   * Обновляем список ребер: Добавляем ребро из вершины 5: (5, 2) с весом 3.
   * Выбираем ребро (5, 2) с весом 3.
   * Добавляем ребро (5, 2) к остову.
   * Обновляем список ребер: Добавляем ребро из вершины 4: (4, 6) с весом 9.
   * Выбираем ребро (4, 6) с весом 9.
   * Добавляем ребро (4, 6) к остову.
   * Обновляем список ребер: Добавляем ребро из вершины 6: (6, 7) с весом 2.
   * Выбираем ребро (6, 7) с весом 2.
   * Добавляем ребро (6, 7) к остову.
   * Обновляем список ребер: Все вершины уже посещены.



(На стрине указанные вершины пронумерованы с 0, а не с 1)

***Задание 7.*** По графу, соответствующему варианту составить минимальное остовное дерево по алгоритму Крускала. Шаги построения отразить в отчете.   
  
Извините за путаницу. Давайте пересчитаем, используя правильную матрицу смежности:

1. \*\*Инициализация\*\*:

- Создаем список всех ребер с их весами:

1 -> 3 len: 1

3 -> 4 len: 7

3 -> 6 len: 11

4 -> 2 len: 4

4 -> 5 len: 1

4 -> 6 len: 6

4 -> 7 len: 9

5 -> 2 len: 3

5 -> 7 len: 10

6 -> 7 len: 2

2. \*\*Сортировка ребер\*\*:

- Отсортируем ребра по их весам по возрастанию:

1 -> 3 len: 1

4 -> 5 len: 1

6 -> 7 len: 2

5 -> 2 len: 3

4 -> 2 len: 4

4 -> 6 len: 6

3 -> 4 len: 7

4 -> 7 len: 9

5 -> 7 len: 10

3 -> 6 len: 11

3. \*\*Присоединение ребер\*\*:

- Начинаем присоединять ребра к остову в порядке возрастания их весов.

4. \*\*Шаги алгоритма\*\*:

- Выбираем ребро (1, 3) с весом 1. Добавляем его к остову.

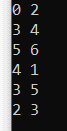
- Выбираем ребро (4, 5) с весом 1. Добавляем его к остову.

- Выбираем ребро (6, 7) с весом 2. Добавляем его к остову.

- Выбираем ребро (5, 2) с весом 3. Добавляем его к остову.

- Выбираем ребро (4, 6) с весом 6. Добавляем его к остову.

- Выбираем ребро (3, 4) с весом 7. Добавляем его к остову.



(На скрине указанные вершины пронумерованы с 0, а не с 1)

Вес будет равен 20

**Вывод**: в ходе лабораторной работы я освоил сущность и программную реализацию: а) способов представления графов; б) алгоритмов поиска в ширину и глубину; в) алгоритма топологической сортировки графов и разобрала алгоритм Прима и алгоритм Крускала