Учреждение образования

«Белорусский государственный технологический университет»

**Отчет по лабораторной работе №6**

**АЛГОРИТМЫ НА ГРАФАХ**

Выполнил:

Студент ФИТ 2-7-2

Лешук Дмитрий

**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.*** Ограф для 10-ого варианта выглядит следующим образом – рисунок 1.

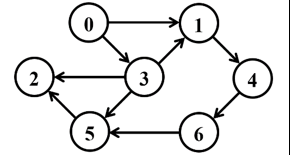


Рисунок 1 – Орграф из условия

Составим матрицу смежности – таблица 1.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **m/n** | **V0** | **V1** | **V2** | **V3** | **V4** | **V5** | **V6** |
| **V0** | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| **V1** | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **V2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **V3** | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| **V4** | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| **V5** | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| **V6** | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Таблица 1 – Матрица смежности

Подпишем рёбра – рисунок 2.

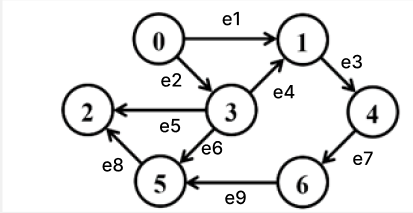


Рисунок 2 – Рёбра графа

Далее составим матрицу инцидентности – таблица 3.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **E1** | **E2** | **E3** | **E4** | **E5** | **E6** | **E7** | **E8** | **E9** |
| **V0** | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **V1** | -1 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 0 |
| **V2** | 0 | 0 | 0 | 0 | -1 | 0 | 0 | -1 | 0 |
| **V3** | 0 | -1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| **V4** | 0 | 0 | -1 | 0 | 0 | 0 | 1 | 0 | 0 |
| **V5** | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 1 | -1 |
| **V6** | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 1 |

Таблица 3 – Матрица инцидентности

Запишем список смежности:

V0: V1, V3

V1: V4

V2:

V3: V2, V5, V1

V4: V6

V5: V2

V6: V5

***Задание 2.***

Продублируем матрицу смежности – таблица 4.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **m/n** | **V0** | **V1** | **V2** | **V3** | **V4** | **V5** | **V6** |
| **V0** | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| **V1** | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **V2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **V3** | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| **V4** | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| **V5** | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| **V6** | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Таблица 4 – Матрица смежности

Сначала выполним обход графа, используя поиск в глубину. Решение:

1. Берём вершину V0 как стартовую. Смотрим в строке для V0, с какой вершиной у графа первое соединение – это V1. Переходим в вершину V1.
2. Повторяем это же действие – переходим в вершину V4.
3. Из V4 – в V6.
4. Из V6 – в V5.
5. Из V5 – во V2.
6. Из V2 у нас не выходит рёбер, однако нам нужно попасть ещё в V3.
7. Возвращаемся в V5, возвращаемся в V6, возвращаемся в V4, в V1, в V0.
8. Из V0 – в V3

Результаты поиска в глубину представлены на рисунке 3. Поиск в глубину аналогичен стеку.

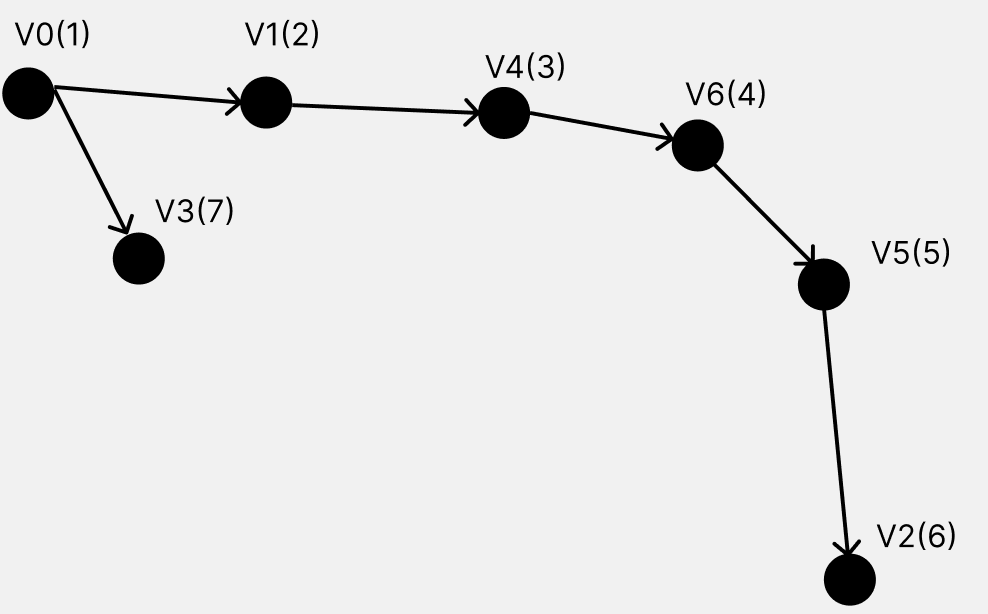


Рисунок 3 – Результат поиска в глубину

Теперь – поиск в ширину. Алгоритм:

1. Берем V0 как стартовую вершину. Это вершина 0-ого уровня.
2. В матрице смежности смотрим, с какими вершинами имеются связи у V0. Это V1 и V3 – вершины первого уровня.
3. Аналогично делаем для V1 и V3. Вершины V4,V2, V5 – вершины второго уровня
4. Вершина V6 – вершина 3-его уровня

Результат поиска в ширину – рисунок 4. Поиск в ширину схож с очередью.

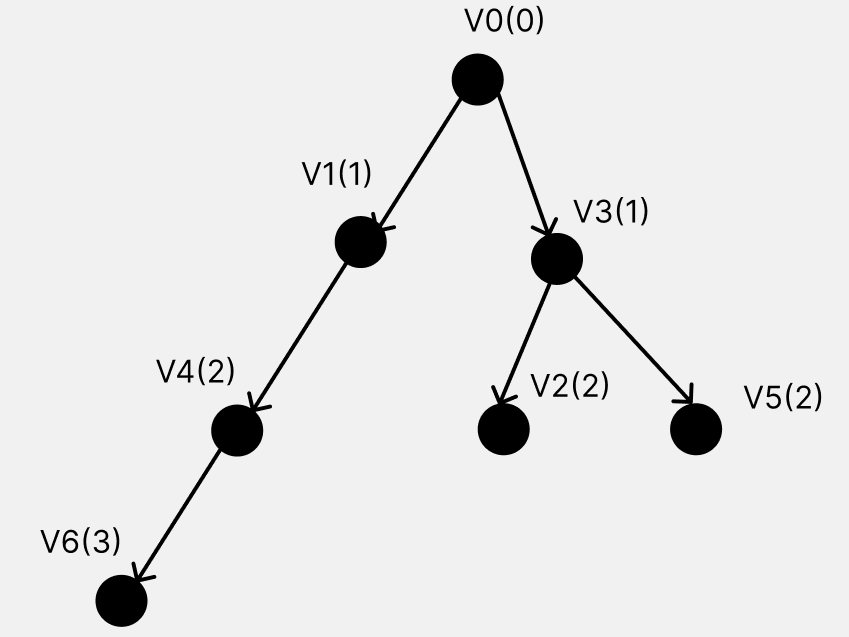


Рисунок 4 – Результат поиска в ширину

Топологическая сортировка

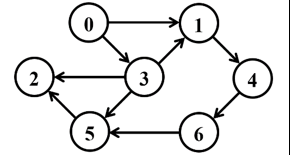


Рисунок 5 – Орграф для топологической сортировки

1. Выбираем вершину, которая не имеет входящих дуг – это вершина 0.  
   Убираем все входящие в вершину 0 дуги.
2. Снова выбираем вершину без исходящих дуг. Это вершина 3. Снова убираем для неё все входящие дуги.
3. Следующая вершина – 1.
4. Таким же образом выбираем 4 вершину
5. Далее – 6.
6. Следующая – 5.
7. Последняя – вершина 2.

Результирующая последовательность топологической сортировки:

**0 – 3 – 1 – 4 – 6 – 5 – 2.**

***Задание 3***

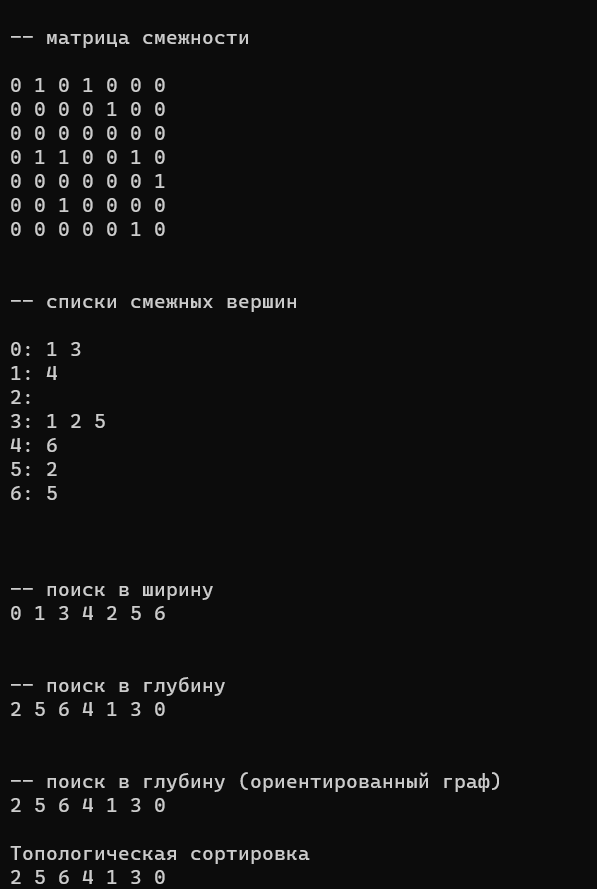


Рисунок 5 – Результат выполнения кода

**Задание 4.**

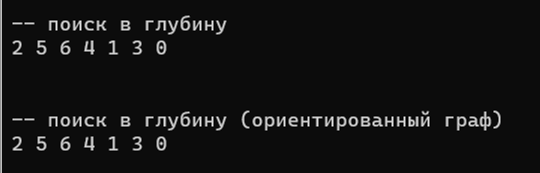
****

Рисунок 6 – Поиск в глубину

**Задание 5.**

Почему-то получилось в обратном порядке.

****

Рисунок 7 – Топологическая сортировка

**Задание 6.**

Алгоритм Прима

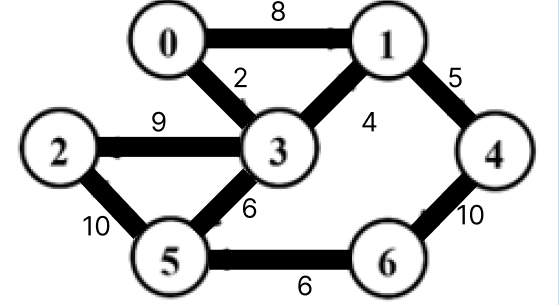


Рисунок 8 – Условие

Ход решения:  
1) Выбираем случайную вершину. Пусть это будет 0. Выбираем ребро с минимальным весом, исходящее из неё. V0 – V3. Вес = 2.  
2) Аналогичное проделываем со всеми последующими вершинами. Из V3 – в V1. Вес этого шага = 4. Итоговый – 6.  
3) V1-V4. Вес = 5. Итоговый – 11.  
4) V4-V6. Вес = 10. Итоговый – 21.  
5) из V6 в V5. Вес = 6. Итоговый = 27.  
6) из V5 в V2. Вес = 10. Итоговый = 37.

Результирующая последовательность: **0 – 3 – 1 – 4 – 6 – 5 – 2.**

**Задание 7.**

Алгоритм Краскала.

Отсортируем ребра в порядке убывания веса:

W(e0,3)=2  
W(e3,1)=4  
W(e1,4)=5

W(e3,5)=6

W(e6,5)=6

W(e0,1)=8

W(e3,2)=9

W(e5,2)=10

W(e4,6)=10

Выбираем 0 – 3. 3 – 1. 1 – 4. 4 – 6. 6 – 5. 5 – 2. В итоге получаем вес 37.

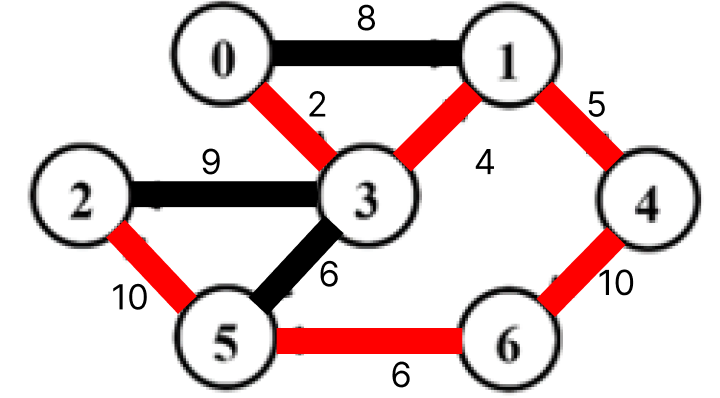


Рисунок 9 – Решение

**Вывод:**В результате лабораторной были освоены сущность и программная реализация способов представления графов; алгоритмов поиска в ширину и глубину; алгоритма топологической сортировки графов. Разобраны алгоритмы Прима и Крускала.