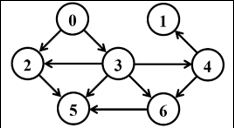
**Лабораторная работа №6**

**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.

**ВАРИАНТ 4**



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

**Матрица смежности \*:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

**Матрица инцидентности:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | I1 | I2 | I3 | I4 | I5 | I6 | I7 | ­I8 | I9 | I10 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | -1 | 0 | 0 | -1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | -1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 | 0 | -1 | 0 | 0 | 0 | 1 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | 0 | 0 | -1 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | 1 |

Список смежных **вершин**:

0: 2, 3 4: 1,6

1: - 5: -

2: 5 6: 5

3: 2,4, 5, 6

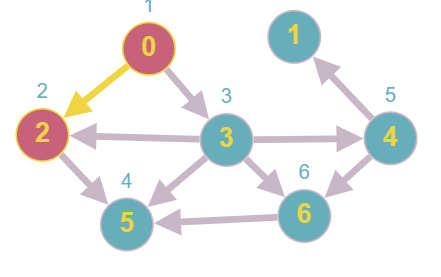
**Задание 2.** Осуществить алгоритмы поиска в ширину и глубину, а также алгоритма топологической сортировки аналогично примерам, рассмотренным на лекциях. Оформить отчет, включив в него каждый шаг выполнения алгоритмов.

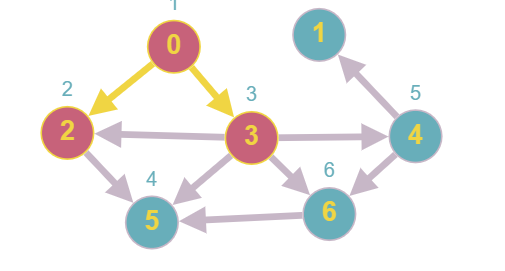
**Поиск в ширину:**

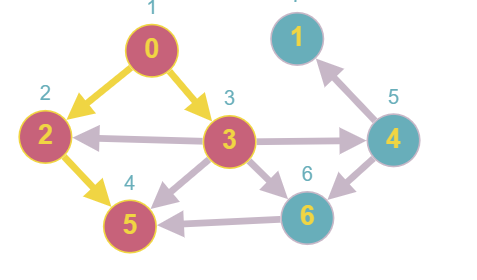
Алгоритм подразумевает, что задана исходная (стартовой) вершина, и основывается на простом правиле: при выборе очередной вершины предпочтение отдается ближайшей.

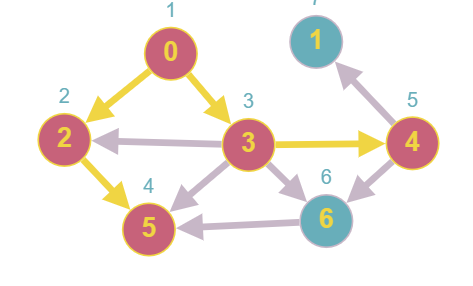
При этом считается, что все дуги графа имеют единичную длину.

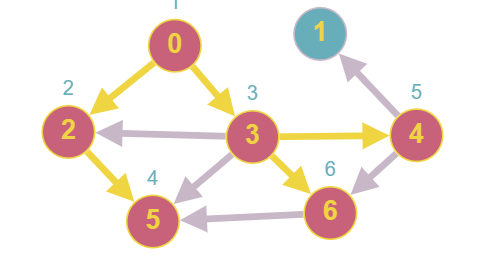
Сначала посещается стартовая вершина, затем все вершины, смежные ей (т. е. находящиеся на расстоянии 1), после чего вершины, находящиеся на расстоянии 2 от стартовой и т.д.

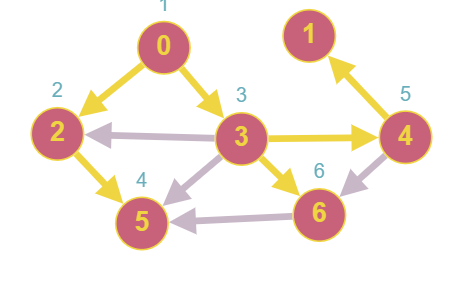
****

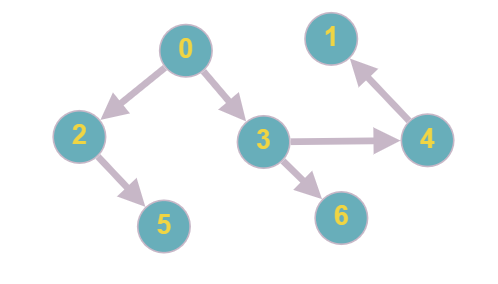








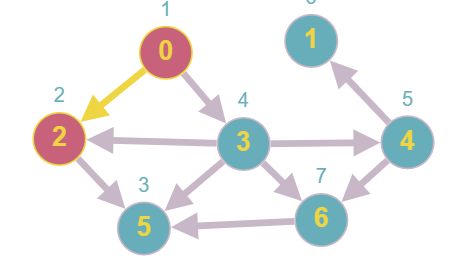
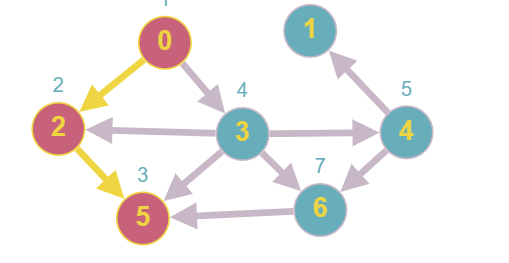


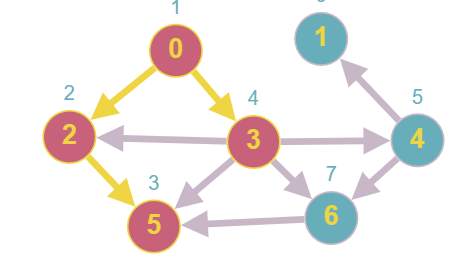


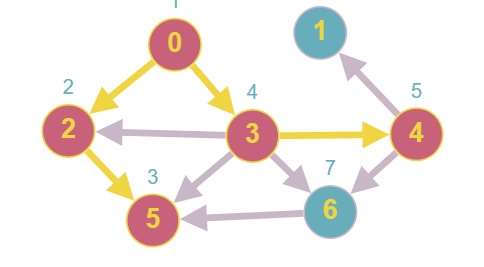
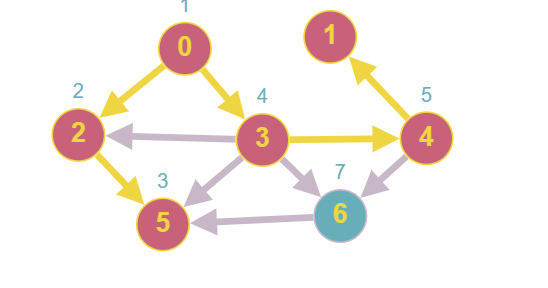
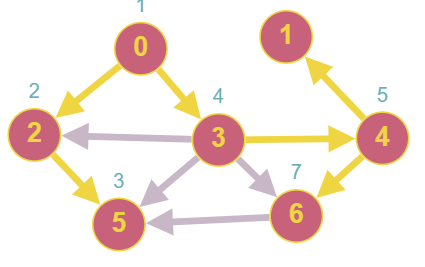
Порядок обхода: **0 2 3 5 4 6 1**

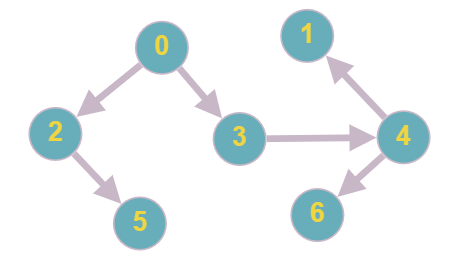
**Поиск в глубину:**

Как и для поиска в ширину, задается стартовая вершина. Алгоритм описывается следующим образом: для каждой не пройденной вершины, начиная со стартовой, необходимо найти все смежные вершины и повторить поиск для каждой.





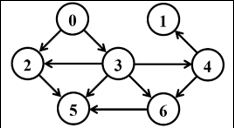
Порядок обхода: **0 2 5 3 4 1 6**

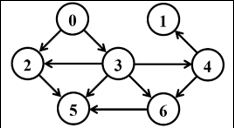
**Топологическая сортировка вершин графа:**

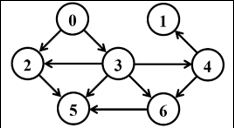
Алгоритм Демукрона вычисления порядковой функции сети:

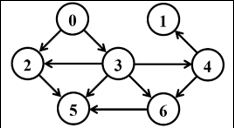
Этот метод состоит в вычислении порядковой функции сети и известен как алгоритм Демукрона. Предполагается, что вершины сети пронумерованы от 1 до n.

Наглядно процесс определения уровней вершин можно представить следующим образом. Нулевой уровень образуют входы сети — вершины с полустепенью захода, равной 0. Удалив из сети все вершины нулевого уровня и исходящие из них дуги, вновь получим сеть, входами которой будут вершины первого уровня исходной сети. Указанный процесс "послойного" удаления вершин следует продолжать до тех пор, пока все вершины исходной сети не будут распределены по уровням.

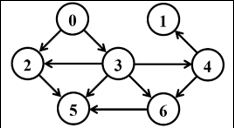




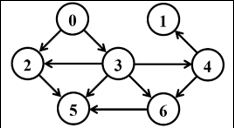




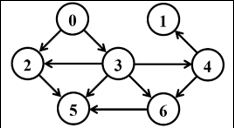
**5**



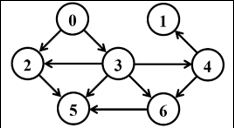
**5 2**



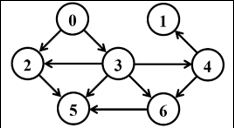
**5 2**



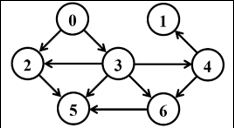
**5 2**



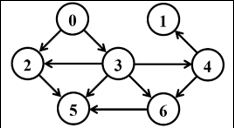
**5 2**



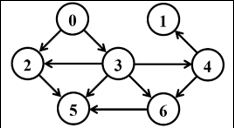
**5 2**



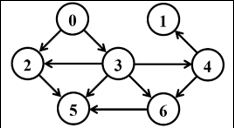
**5 2 1**



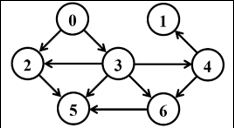
**5 2 1**



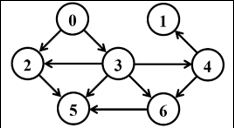
**5 2 1 6**



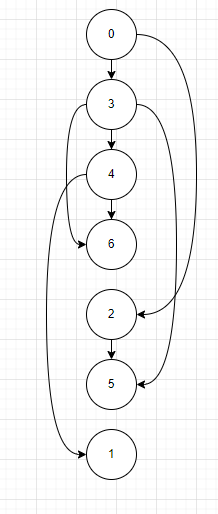
**5 2 1 6 4**



**5 2 1 6 4 3**



**5 2 1 6 4 3 0**

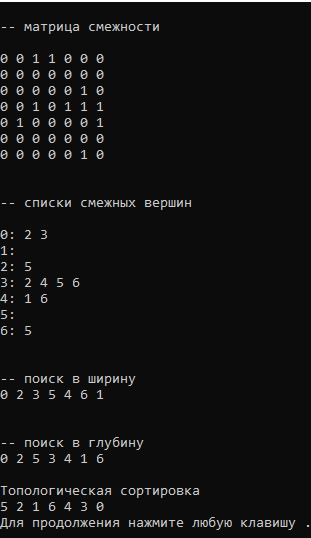


Топологически сортированный граф.

Задание 3. 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 4. Разработать функцию DFS обхода вершин графа, используя метод поиска глубину. Продемонстрировать работу функции. Копии экрана вставить в отчет.

Задание 5. Доработайте функцию DFS, для выполнения топологической сортировки графа. Продемонстрировать работу функции. Копии экрана вставить в отчет.



**Задание 6.** По графу, соответствующему варианту составить минимальное остовное дерево по алгоритму Прима. Шаги построения отразить в отчете.

Веса ребер принять:

W:

W(e0,1)=8; W(e1,0)=5;

W(e0,2)=1; W(e2,0)=3;

W(e0,3)=2; W(e3,0)=8;

W(e1,3)=11; W(e3,1)=4;

W(e1,4)=5; W(e4,1)=3;

W(e2,3)=7; W(e3,2)=9;

W(e2,5)=11; W(e5,2)=10;

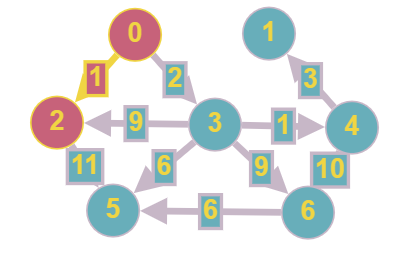
W(e4,3)=4; W(e3,4)=1;

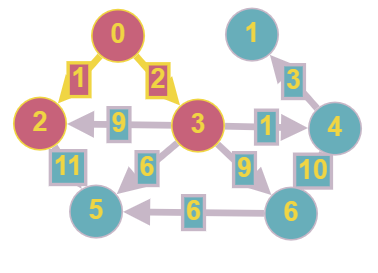
W(e4,6)=10; W(e6,4)=2;

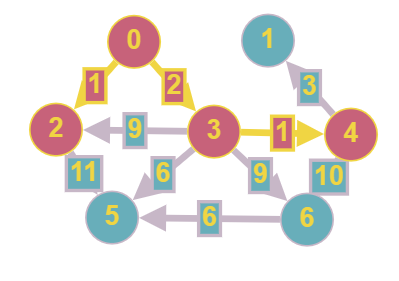
W(e5,6)=2; W(e6,5)=6;

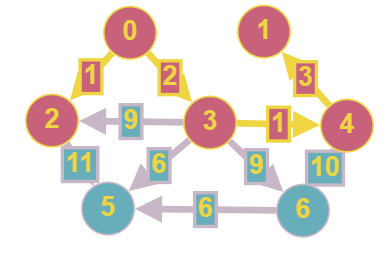
W(e5,3)=3; W(e3,5)=6;

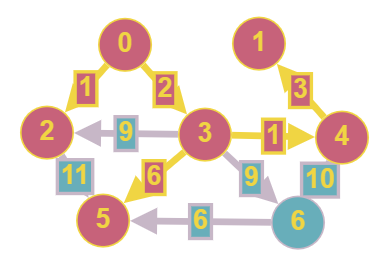
W(e6,3)=7; W(e3,6)=9;

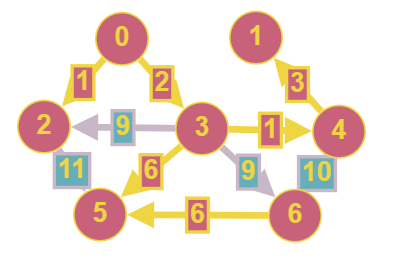


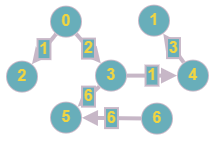


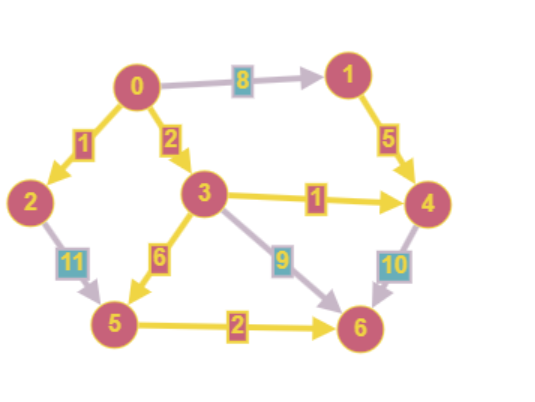










**Вес минимального остовного дерева равен 19**

**Задание 7.** По графу, соответствующему варианту составить минимальное остовное дерево по алгоритму Крускала. Шаги построения отразить в отчете.

W(e0,2)=1; W(e3,4)=1;

W(e5,6)=2; W(e0,3)=2; W(e6,4)=2;

W(e5,3)=3; W(e2,0)=3; W(e4,1)=3;

W(e1,0)=5; W(e1,4)=5;

W(e0,1)=8; W(e3,0)=8;

W(e2,3)=7; W(e6,3)=7;

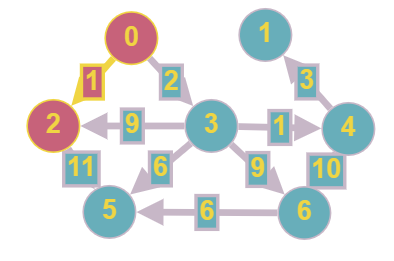
W(e3,1)=4; W(e4,3)=4;

W(e6,5)=6; W(e3,5)=6;

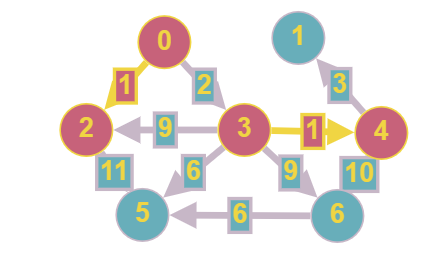
W(e3,2)=9; W(e3,6)=9;

W(e2,5)=11; W(e1,3)=11;

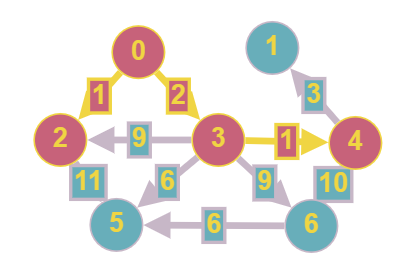
W(e4,6)=10; W(e5,2)=10;



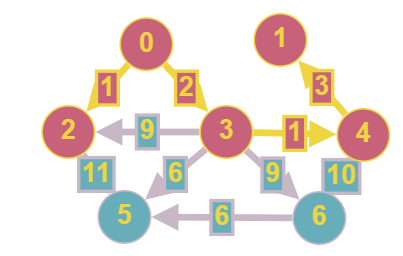
(e0, 2)



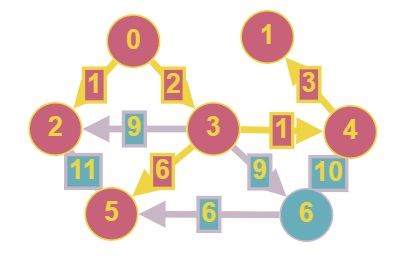
(e0, 2), (e3, 4),



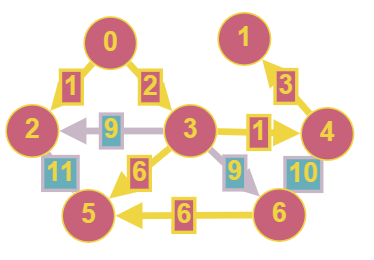
(e0, 2), (e3, 4), (e0, 3)



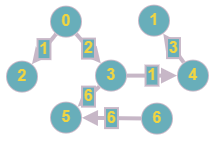
(e0, 2), (e3, 4), (e0, 3), (e4, 1)



(e0, 2), (e3, 4), (e0, 3), (e4, 1), (e3, 5)



(e0, 2), (e3, 4), (e0, 3), (e4, 1), (e3, 5), (e6, 5)



**Вывод:** По результатам лабораторной работы мы освоили 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.