

ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова»

Кафедра вычислительной техники

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

Вариант 7

Выполнил: студент группы ИВТ-41-20

Галкин Дмитрий Сергеевич

Проверил:

доцент

Павлов Леонид Александрович

Чебоксары

2022

**Цель работы:** ознакомление с алгоритмами на графах, получение практических навыков программирования задач, которые формулируются в терминах теории графов.

1. Разработать алгоритмы и программы ввода и вывода произвольного графа.

2. Разработать алгоритмы и программы перехода от одного представления графа к другому в соответствии с вариантами, заданными в табл.; оценить вычислительную сложность разработанных алгоритмов.

**Таблица**

| № варианта | 1 | 2 | 3 | 4 |
|------------|---|---|---|---|
| 1          | + | + |   |   |
| 2          | + |   | + |   |
| 3          | + |   |   | + |
| 4          |   | + | + |   |
| 5          |   | + |   | + |
| 6          |   |   | + | + |

Номера столбцов соответствуют следующим представлениям графа:

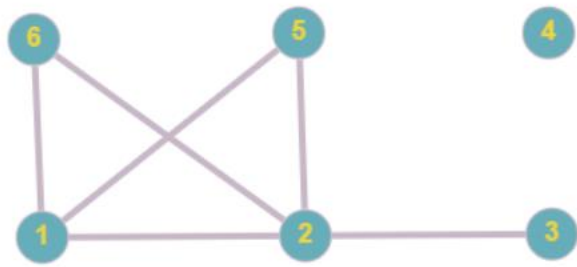
1. Матрица смежности.
2. Матрица инцидентий.
3. Список ребер.
4. Структура смежности.

3. Реализовать алгоритм Краскала нахождения минимального остовного дерева взвешенного графа.

4. Реализовать алгоритм Дейкстры-Прима нахождения минимального остовного дерева взвешенного графа.

5. Реализовать метод построения фундаментального множества циклов неориентированного графа.

6. Реализовать поиск кратчайших путей от фиксированной вершины до всех остальных вершин орграфа.

|   |   |
|---|---|
|  | <pre>       1 2 3 4 5 6 1 (0, 1, 0, 0, 1, 1) 2 (1, 0, 1, 0, 1, 1) 3 (0, 1, 0, 0, 0, 0) 4 (0, 0, 0, 0, 0, 0) 5 (1, 1, 0, 0, 0, 0) 6 (1, 1, 0, 0, 0, 0) </pre> <p>Матрица смежности</p> |
|---|---|

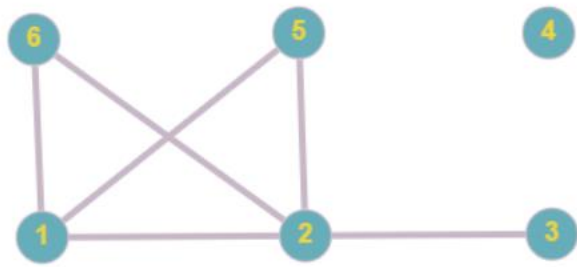
### Переход от матрицы смежности к матрицу инцидентий

```

n // количество ребер
size // количество вершин
a[size][size] // матрица смежности
i[size][n] // матрица инцидентий
n := 0
for i := 0 to size) {
  for j := i + 1 to size) {
    if (a[i][j] == 1) {
      i[i][n] := 1;
      i[j][n] := 1;
      Inc(n);
    }
  }
}

```

|   |   |
|---|---|
| <pre>       1 2 3 4 5 6 1 (0, 1, 0, 0, 1, 1) 2 (1, 0, 1, 0, 1, 1) 3 (0, 1, 0, 0, 0, 0) 4 (0, 0, 0, 0, 0, 0) 5 (1, 1, 0, 0, 0, 0) 6 (1, 1, 0, 0, 0, 0) </pre> <p>Матрица смежности</p> | <pre> p  1 2 3 4 5 6 1 (1, 1, 1, 0, 0, 0) 2 (1, 0, 0, 1, 1, 1) 3 (0, 0, 0, 1, 0, 0) 4 (0, 0, 0, 0, 0, 0) 5 (0, 1, 0, 0, 1, 0) 6 (0, 0, 1, 0, 0, 1) </pre> <p>Матрица инцидентий</p> |
|---|---|

|   |   |
|---|---|
|  | <pre> p  1 2 3 4 5 6 1 (1, 1, 1, 0, 0, 0) 2 (1, 0, 0, 1, 1, 1) 3 (0, 0, 0, 1, 0, 0) 4 (0, 0, 0, 0, 0, 0) 5 (0, 1, 0, 0, 1, 0) 6 (0, 0, 1, 0, 0, 1) </pre> <p>Матрица инцидентий</p> |
|---|---|

### Переход от списка ребер к матрице инцидентий:

```

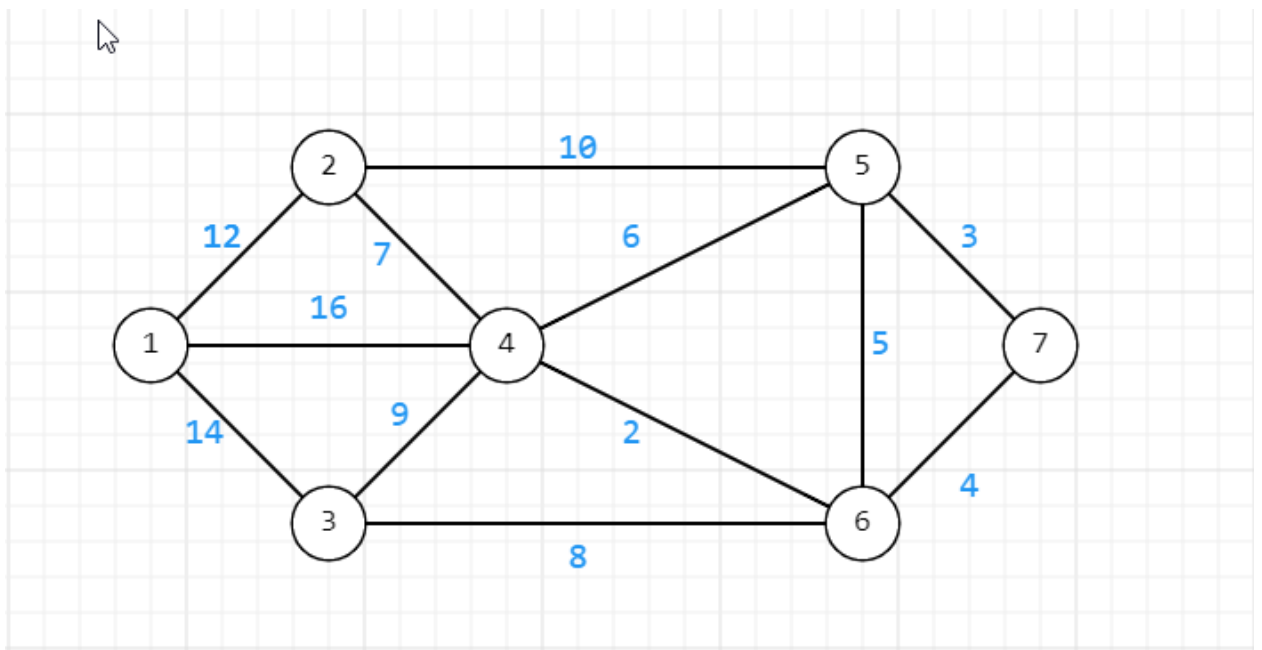
col // количество ребер
row // количество вершин
i[row][col] // матрица инцидентий
a[row][row] // матрица смежности
for i := 0 to row {
  for j := 0 to col {
    if (i[i][j] == 1) {
      for i1 := i + 1 to row {
        if (i[i1][j] == 1) {
          a[i1][i] := 1;
          a[i][i1] := 1;
          break;
        }
      }
    }
  }
}

```

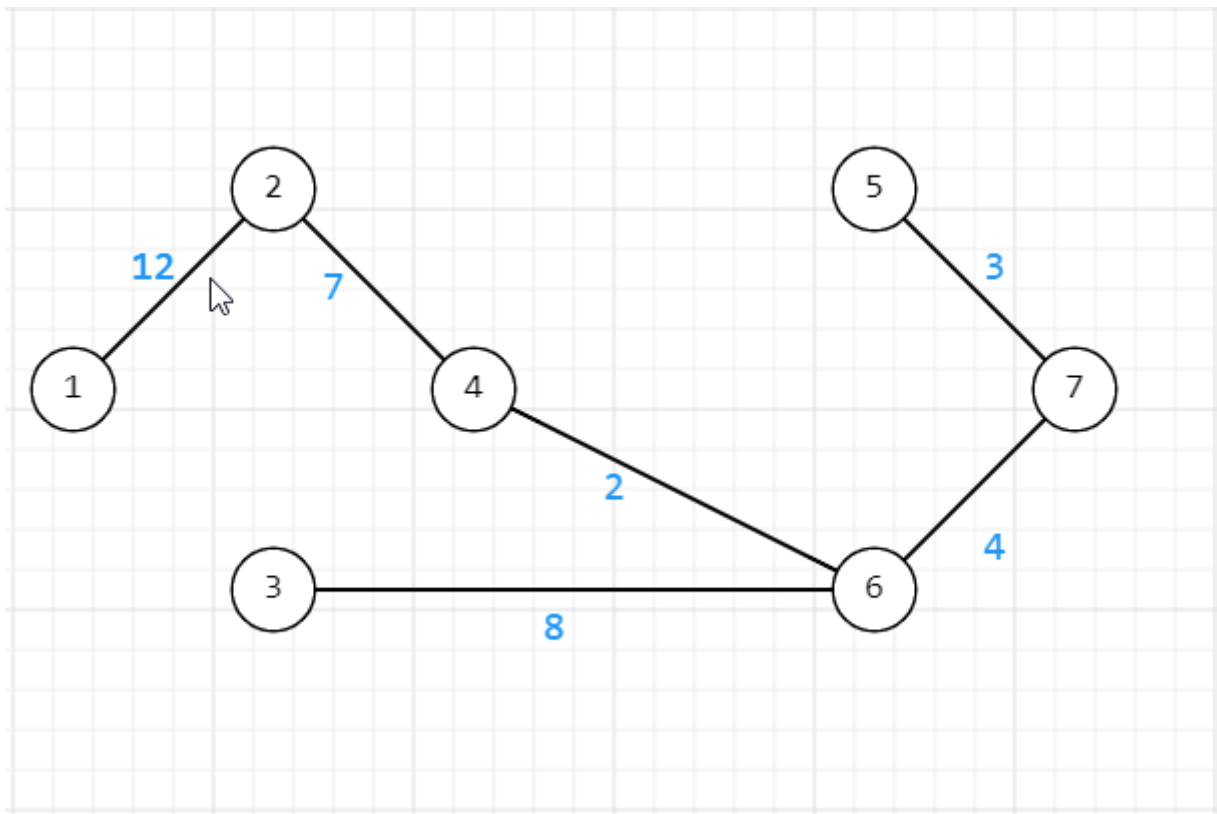
|   |   |
|---|---|
| <pre> p  1 2 3 4 5 6 1 (1, 1, 1, 0, 0, 0) 2 (1, 0, 0, 1, 1, 1) 3 (0, 0, 0, 1, 0, 0) 4 (0, 0, 0, 0, 0, 0) 5 (0, 1, 0, 0, 1, 0) 6 (0, 0, 1, 0, 0, 1) </pre> <p>Матрица инцидентий</p> | <pre> 1 2 3 4 5 6 1 (0, 1, 0, 0, 1, 1) 2 (1, 0, 1, 0, 1, 1) 3 (0, 1, 0, 0, 0, 0) 4 (0, 0, 0, 0, 0, 0) 5 (1, 1, 0, 0, 0, 0) 6 (1, 1, 0, 0, 0, 0) </pre> <p>Матрица смежности</p> |
|---|---|

| Этап 1                             | Этап 2                         |
|------------------------------------|--------------------------------|
| Матрица смежности пустая: null     | Матрица смежности пустая: null |
| Матрица инцидентности пустая: null | Матрица инцидентности:         |
| Матрица смежности:                 | 1 1 1 0 0 0                    |
| 0 1 0 0 1 1                        | 1 0 0 1 1 1                    |
| 1 0 1 0 1 1                        | 0 0 0 1 0 0                    |
| 0 1 0 0 0 0                        | 0 0 0 0 0 0                    |
| 0 0 0 0 0 0                        | 0 1 0 0 1 0                    |
| 1 1 0 0 0 0                        | 0 0 1 0 0 1                    |
| 1 1 0 0 0 0                        | Перевод в матрицу смежности:   |
| Перевод в матрицу инцидентности:   | 0 1 0 0 1 1                    |
| 1 1 1 0 0 0                        | 1 0 1 0 1 1                    |
| 1 0 0 1 1 1                        | 0 1 0 0 0 0                    |
| 0 0 0 1 0 0                        | 0 0 0 0 0 0                    |
| 0 0 0 0 0 0                        | 1 1 0 0 0 0                    |
| 0 1 0 0 1 0                        | 1 1 0 0 0 0                    |
| 0 0 1 0 0 1                        |                                |

## Неоринтированный граф



## Минимальное остовное дерево



-----  
Алгоритм Крускала нахождение минимального остовного дерева взвешенного графа.

4 - 6 = 2  
5 - 7 = 3  
6 - 7 = 4  
2 - 4 = 7  
3 - 6 = 8  
1 - 2 = 12

The weight = 36

-----  
Алгоритм Дейкстры-Прима нахождение минимального остовного дерева взвешенного графа.

3 - 6 = 8  
4 - 6 = 2  
6 - 7 = 4  
5 - 7 = 3  
2 - 4 = 7  
1 - 2 = 12

The weight = 36

Реализовать метод построения фундаментального множества циклов неориентированного графа.

6->4->2->1->3->6->5->7->6

3->4->5->6->3

1->2->4->6->3->1

7->5->2->1->3->4->6->7

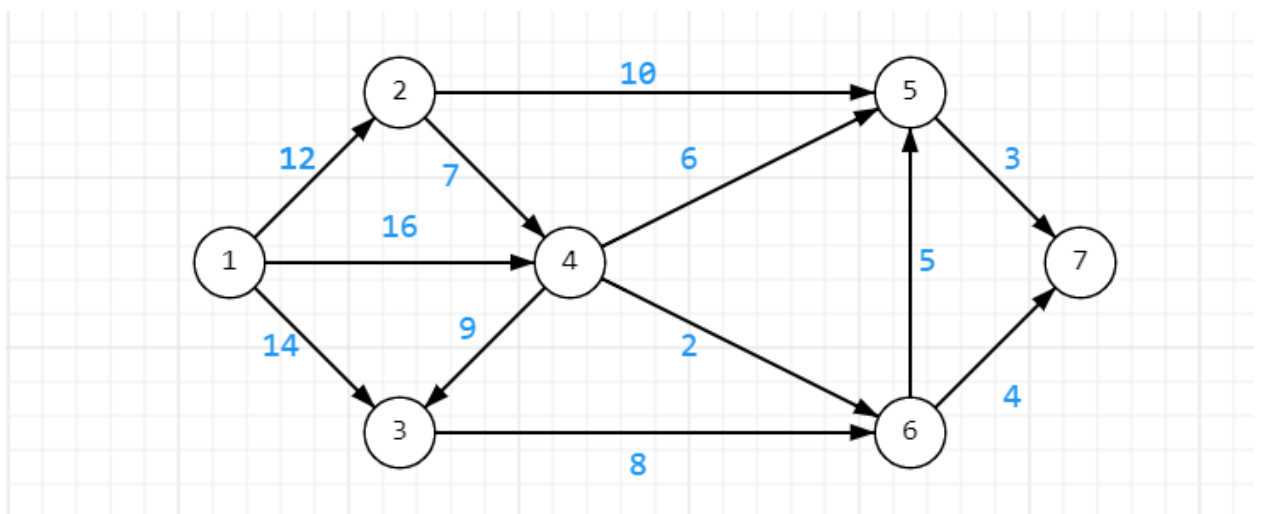
4->2->5->6->4

4->1->2->5->4->6->3->4

4->1->2->5->7->6->4

4->5->6->3->4->1->2->4

## Ориентированный граф



Реализовать поиск кратчайших путей от фиксированной вершины до всех остальных вершин орграфа.

1 -> 2 = 12

1 -> 3 = 14

1 -> 4 = 16

1 -> 5 = 22

1 -> 6 = 18

1 -> 7 = 22

**Вывод:** ознакомился с алгоритмами на графах, получил практические навыки программирования задач, которые формулируются в терминах теории графов.

