

# Лабораторные работы по дисциплине «Теория графов»

Ali Shadow

30 сентября 2019 г.

# Содержание

ЛР № 1. Основная программа.	4
ЛР № 2. Поиск пути.	7
ЛР № 3. Алгоритм Дейкстры и близкие к нему.	8
ЛР № 4. Эксцентриситет, радиус, диаметр и степени.	9
ЛР № 5. Изоморфизм графов.	10
ЛР № 6. Связность графа.	11
ЛР № 7. Дополнение для графа.	12
ЛР № 8. Бинарные операции над графами.	13
ЛР № 9. Планарность графа.	14
ЛР № 10. Восстановление графа из вектора.	15
ЛР № 11. Экстремальные графы.	16
ЛР № 12. Деревья и минимальное остовное дерево.	17
ЛР № 13. Задача о цикле.	18
ЛР № 14. Раскраска графа.	19
ЛР № 15. Задача о свадьбах.	20
ЛР № 16. Дополнение 2-комплекса.	21

ЛР № 17. Восстановление 2-комплекса из вектора.	22
ЛР № 18. Экстремальные 2-комплексы.	23
ЛР № 19. Задача коммивояжёра.	24
ЛР № 20. Доминирование и покрывающее множество.	25
ЛР № 21. Эффективность алгоритмов поиска пути.	26
Список литературы	28

# ЛР № 1. Основная программа.

Написать программу, которая имеет следующие возможности.

1. Может принимать на вход:

- (a) матрицу смежности графа;
- (b) матрицу инцидентности графа;
- (c) список рёбер вершин вида  $Edges\{i(a, k, l, d), \dots\}$ , где  $i$  — номер ребра,  $a$  — вес ребра,  $k$  и  $l$  — номера или имена вершин,  $d$  — может быть 1, если ребро направленное, и 1 в другом случае;
- (d) список вершин также может быть задан с привязкой по координатам в следующем виде:  $Vertex\{v(x, y)\}$ , где  $v$  — имя или номер вершины,  $x$  и  $y$  — координаты в пикселях;
- (e) если список вершин не указан, то их координаты могут задаваться произвольно;
- (f) каждый тип входных файлов имеет своё расширение и свою структуру;
- (g) в файле могут быть комментарии начинающиеся с %.

2. Позволяет задать граф графическим способом, с помощью мыши и клавиатуры:

- (a) можно устанавливать вершины мышью;
- (b) можно мышью связать вершины между собой с указанием веса на на клавиатуре или с помощью элементов управления;
- (c) можно выбирать направление рёбер или оставлять их ненаправленными через контекстное меню у рёбер;
- (d) можно менять форму рёбер с помощью мыши;

- (e) можно делать петли, через контекстное меню у вершин;
  - (f) можно мышью двигать вершины, за ними соответственно передвигаются инцидентные им рёбра;
  - (g) можно строить мультиграф, то есть несколько рёбер, между парами вершин, при этом рёбра не должны накладываться друг на друга.
3. Каждый новый граф открывается в отдельной вкладке основной части окна.
  4. В процессе работы сохраняются 10 последних состояний каждого графа, также в интерфейсе есть соответствующие функции «Undo» и «Redo».
  5. Позволяет сохранить полученный граф в тех же форматах, что и входные данные, а также в виде изображения.
  6. Показывает в реальном времени матрицу смежности графа.
  7. При изменении матрицы смежности графа сразу меняются рёбра графа (вес или наличие ребра).
  8. У приложения должен быть графический интерфейс:
    - (a) должно быть меню с пунктами «Файл», «Задачи теории графов», «?»;
    - (b) в пункте файл должны быть пункты открытия и графа из файла, сохранения графа в файла и выход из программы;
    - (c) для открытия и закрытия файла используются стандартные диалоги;

(d) если пользователь редактировал или создал граф и не сохранил его, то перед выходом программа должна предложить сохранить граф в файл;

(e) в пункте меню должны быть следующие пункты:

- «О программе» с исчерпывающими инструкциями по использованию вашей программой, в том числе информация по форматам входных файлов;
- «Об авторе» с ФИО автора (авторов для первой лабораторной работы), его группе, кафедре, факультете и университете.

9. Дальнейшие лабораторные работы используют инфраструктуру данной лабораторной и добавляются в пункты её меню, в раздел «Задачи теории графов».

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 2. Поиск пути.

Добавить в основную программу следующий функционал:

1. по выбранному графу (полученному любым из предложенных в основной программе вариантов) предлагается выбрать метод поиска пути (который впоследствии реализуется)
  - (a) **Breadth-First Search** для пары указанных вершин;
  - (b) **A\*** для пары указанных вершин;
  - (c) **Iterative deepening A\*** для пары указанных вершин;
2. перед реализацией графа, он проверяется на требования, предъявляемые алгоритмом к графу, а при несоблюдении оных сообщается об этом;
3. можно посмотреть расстояние между указанными вершинами;
4. на графе отображается один из кратчайших маршрутов;
5. полученный маршрут можно сохранить в файл.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)



## ЛР № 3. Алгоритм Дейкстры и близкие к нему.

Добавить в основную программу следующий функционал:

1. по выбранному графу (полученному любым из предложенных в основной программе вариантов) предлагается выбрать метод поиска пути (который впоследствии реализуется)
  - (а) **Флойда-Уоршелла** для указанной вершины;
  - (b) **Дейкстры** для указанной вершины;
  - (с) **Беллмана-Форда** для указанной вершины;
  - (d) **Джонсона** для указанной вершины;
2. перед реализацией графа, он проверяется на требования, предъявляемые алгоритмом к графу, а при несоблюдении оных сообщается об этом;
3. можно посмотреть вектор расстояний от указанной вершины до остальных;
4. можно посмотреть матрицу расстояний от каждой вершины до каждой;
5. вектор и матрицу можно сохранить в файл.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)



## ЛР № 4. Эксцентриситет, радиус, диаметр и степени.

Добавить в основную программу следующий функционал для выбранному графа (полученного любым из предложенных в основной программе вариантов):

1. находятся все веса вершин (для нахождения можно воспользоваться любой из лабораторных: [2](#) или [3](#));
2. находится радиус графа;
3. находится диаметр графа;
4. строится вектор степеней вершин графа;
5. все найденные показатели можно сохранить в файл.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 5. Изоморфизм графов.

Добавить в основную программу следующий функционал:

1. на вход принимаются два графа (полученные любым из предложенных в основной программе вариантов), определяется их изоморфность:
  - (а) если вектора степеней вершин не равны друг друга, то сообщаем о сильной неизоморфности;
  - (b) иначе проверяем изоморфность (алгоритм Ульмана) и сообщаем о результатах

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 6. Связность графа.

Добавить в основную программу следующий функционал по выбранному графу (полученному любым из предложенных в основной программе вариантов):

1. определяется его связность (например с помощью ЛР [3](#));
2. если граф является ориентированным, определить, является ли он слабо-связным, сильно-связным или несвязным;
3. определяется количество компонент графа и их состав;
4. определяется количество мостов в графе;
5. определяется количество шарниров в графе.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 7. Дополнение для графа.

Добавить в основную программу следующий функционал по выбранному графу (полученному любым из предложенных в основной программе вариантов):

1. определяется его полнота, если он полный, то сообщается;
2. строится дополнение графа до полного;
3. полученный граф можно сохранить.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 8. Бинарные операции над графами.

Добавить в основную программу следующий функционал по выбранной паре графов  $A$  и  $B$  (полученной любым из предложенных в основной программе вариантов):

1. Пользователь выбирает одну из логических функций для двух графов с одинаковым множеством вершин для постройки нового графа:

- (a) объединение  $A$  и  $B$ ;
- (b) пересечение  $A$  и  $B$ ;
- (c) вычитание  $B$  из  $A$ ;
- (d) вычитание  $A$  из  $B$ ;
- (e) импликация из  $A$  в  $B$ ;
- (f) импликация из  $B$  в  $A$ ;
- (g) коимпликация из  $A$  в  $B$ ;
- (h) коимпликация из  $B$  в  $A$ ;
- (i) исключающее или  $A$  и  $B$ ;
- (j) эквивалентность  $A$  и  $B$ ;
- (k) штрих Шеффера  $A$  и  $B$ ;
- (l) стрелка Пирса  $A$  и  $B$ ;

2. веса рёбер не имеют значения и в результирующем графе равны 1.

**Срок сдачи: 15.12.2019**

[\[К оглавлению\]](#)

## ЛР № 9. Планарность графа.

Добавить в основную программу следующий функционал по выбранному графу (полученному любым из предложенных в основной программе вариантов):

1. определить, является ли граф плоским;
2. определить, является ли граф планарным;
3. если граф является планарным, построить для него двойственный граф, с возможностью сохранить полученный граф в файл.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 10. Восстановление графа из вектора.

Добавить в основную программу следующий функционал:

1. на вход принимается вектор степеней вершин;
2. последовательно выполняется редукционный алгоритм;
3. печатаются промежуточные и конечный результаты работы редукционного алгоритма;
4. если граф совершенный или экстремальный, то сообщается об этом;
5. если граф экстремальный, то строится его база;
6. рисуется построенный граф, если это возможно;
7. полученный граф можно сохранить в файле.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 11. Экстремальные графы.

Добавить в основную программу следующий функционал:

1. на вход принимается 2 базы в виде наборов двоек;
2. проверяется, что базы заданы верно;
3. для каждой из баз строится граф и вектор степеней вершин, которые база задаёт;
4. из полученных графов строятся следующие графы:
  - (a) дополнение первого графа;
  - (b) дополнение второго графа;
  - (c) объединение двух графа;
  - (d) пересечение двух графов;
  - (e) вычитание из первого графа второго графа;
  - (f) вычитание из второго графа первого графа.
5. для полученных графов выводятся:
  - (a) список рёбер;
  - (b) векторы степеней вершин;
  - (c) базы при экстремальности соответствующих графов.
6. все полученные графы можно сохранить в файлах.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)



## ЛР № 12. Деревья и минимальное остовное дерево.

Добавить в основную программу следующий функционал по выбранному графу (полученному любым из предложенных в основной программе вариантов):

1. выбирается алгоритм построения остовного дерева:
  - (a) алгоритм Прима;
  - (b) алгоритм Краскала;
  - (c) алгоритм Борувки;
2. определяется возможность на заданном графе реализовать выбранный алгоритм;
3. с помощью выбранного алгоритма строится остовное минимальное дерево;
4. полученный граф можно сохранить в файл.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 13. Задача о цикле.

Добавить в основную программу следующий функционал по выбранному графу (полученному любым из предложенных в основной программе вариантов):

1. определяется наличие циклов;
2. если циклов нет, то находится центр дерева и соответственно его глубина;
3. если циклов нет, сохраните его в виде кода Прюфера, который можно сохранить в файл
4. если циклы есть, то находится минимальный цикл;
5. минимальный цикл отображается на графе;
6. полученное изображение можно сохранить в графический файл.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 14. Раскраска графа.

Добавить в основную программу следующий функционал по выбранному графу (полученному любым из предложенных в основной программе вариантов):

1. определяется его хроматическое число;
2. раскрашиваются вершины графа;
3. полученный граф можно сохранить в виде изображения.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 15. Задача о свадьбах.

Добавить в основную программу следующий функционал по выбранному графу (полученному любым из предложенных в основной программе вариантов):

1. задаются 2 доли графа;
2. проверяется двудольность полученного графа с указанными долями;
3. решается задача о свадьбах;
4. полученный граф можно сохранить в виде изображения.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 16. Дополнение 2-комплекса.

Добавить в основную программу следующий функционал по выбранному 2-комплексу (полученному аналогично предложенным в основной программе вариантам):

1. определяется его полнота, если он полный, то сообщается;
2. строится дополнение 2-комплекса до полного;
3. полученный 2-комплекс можно сохранить в файл.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 17. Восстановление 2-комплекса из вектора.

Добавить в основную программу следующий функционал:

1. на вход принимается вектор степеней;
2. последовательно выполняется редукционный алгоритм для 2-комплекса;
3. печатаются промежуточные и конечный результаты работы редукционного алгоритма;
4. если 2-комплекс совершенный или экстремальный, то сообщается об этом;
5. если 2-комплекс экстремальный, то строится его база;
6. выводятся построенные симплексы, если вектор реализуем;
7. полученный 2-комплекс можно сохранить в файле.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 18. Экстремальные 2-комплексы.

Добавить в основную программу следующий функционал:

1. на вход принимается 2 базы в виде наборов троек;
2. проверяется, что базы заданы верно;
3. для каждой из баз строится набор симплексов и вектор степеней вершин, которые база задаёт;
4. из полученных 2-комплексов строятся следующие 2-комплексы:
  - (a) дополнение первого 2-комплекса;
  - (b) дополнение второго 2-комплекса;
  - (c) объединение двух 2-комплексов;
  - (d) пересечение двух 2-комплексов;
  - (e) вычитание из первого 2-комплекса второго 2-комплекса;
  - (f) вычитание из второго 2-комплекса первого 2-комплекса.
5. для полученных 2-комплексов выводятся:
  - (a) список симплексов;
  - (b) векторы степеней вершин;
  - (c) базы при экстремальности соответствующих комплексов.
6. все полученные 2-комплексы можно сохранить в файлах.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 19. Задача коммивояжёра.

Добавить в основную программу следующий функционал по выбранному графу (полученному любым из предложенных в основной программе вариантов):

1. на вход принимается множество вершин с указанием начала движения;
2. при необходимости определяется конечная вершина движения;
3. с помощью алгоритма **ветвей и границ** решается задача коммивояжёра;
4. выводится конечный маршрут и его длина;
5. результаты могут быть сохранены в файл.

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)



## ЛР № 20. Доминирование и покрывающее множество.

Добавить в основную программу следующий функционал по выбранному графу (полученному любым из предложенных в основной программе вариантов):

1. определите число доминирования выбранного графа;
2. определите мощность наибольшего независимого множества вершин графа;
3. определите мощность наибольшего независимого множества рёбер графа;
4. определите наименьшую мощность покрытия графа;
5. определите наименьшую мощность рёберного покрытия графа;

***Срок сдачи: 15.12.2019***

[\[К оглавлению\]](#)

## ЛР № 21. Эффективность алгоритмов поиска пути.

Добавить в основную программу следующий функционал:

1. генерируется случайный связанный граф ( $\approx 10000$  узлов), вписанный в квадрат;
2. дальние связи крайне редки;
3. веса рёбер соответствуют расстоянию между вершинами;
4. выбираются 4 пары вершин:
  - в серединах левой и правой сторон;
  - в противоположенных углах;
  - в центре квадрата и в середине нижней стороны;
  - произвольно указанные вершины;
5. для каждой пары вершин найти кратчайший маршрут с помощью алгоритмов BFS,  $A^*$ , IDA (ЛР 2), Дейкстры (ЛР 3);
6. кроме маршрута нужно показать все вершины задействованные в ходе работы каждого из алгоритмов;
7. для каждой пары вершин нужны изображения с найденным маршрутом и задействованными вершинами по каждому из алгоритмов;
8. также необходимо привести временные замеры работы алгоритмов для каждой пары вершин;
9. все характеристики можно сохранить в файле.

**Срок сдачи: 15.12.2019**

[\[К оглавлению\]](#)

## Список литературы

1. *Мокряков А. В., Селин П. С., Цурков В. И.* Минимакс и восстановление по вектору в графах. М.: Спутник+, 2017, 308 с.
2. *Оре О.* Теория графов. М.: Наука, 1980, 336 с.
3. *Уилсон Р.* Введение в теорию графов. М.: Мир, 1977, 208 с.
4. *Харари Ф.* Теория графов. М.: Мир, 1973, 300 с.

[\[К оглавлению\]](#)