

Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Сибирский государственный университет телекоммуникаций и
информатики»
(СибГУТИ)

Кафедра Прикладной математики и кибернетики

Расчётно-графическая работа
по теме «Доказательство с нулевым знанием для задачи «Гамильтонов
цикл»»
Вариант 2

Выполнил:
студент группы ИВ-921
Ярошев Р. А.

Проверил:
ассистент кафедры ПМиК
Меркулов И. А.

Новосибирск 2022 г.

Оглавление

Постановка задачи.....	3
Реализация программы.....	4
Результат работы программы.....	5
Листинг.....	6

Постановка задачи

Необходимо написать программу, реализующую протокол доказательства с нулевым знанием для задачи «Гамильтонов цикл».

Задача является NP-полной и не имеет быстрых методов для решения, поэтому для тестирования необходимо будет генерировать правильные решения при помощи дополнительно разработанных программ. Необходимо информацию о графах считывать из файла.

Реализация программы

Была реализована программа, которая реализует протокол доказательства с нулевым знанием для задачи «Гамильтонов цикл» с использованием языка программирования Python.

Функции:

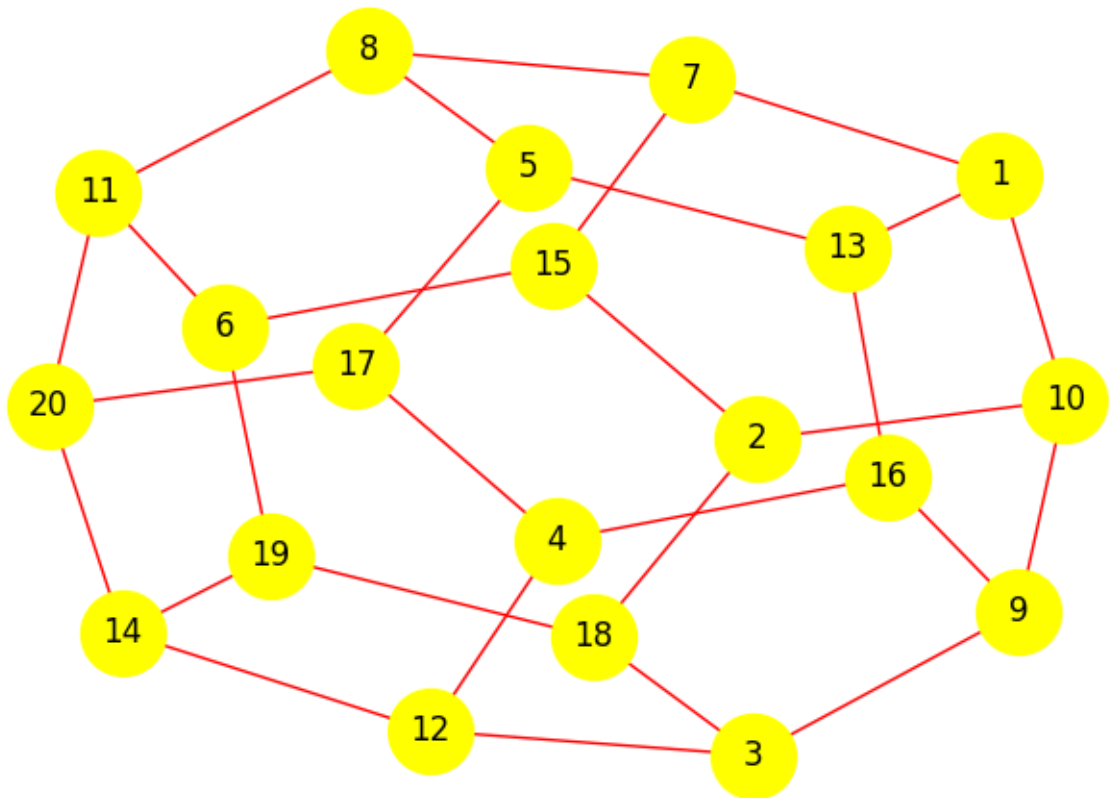
- `graph_drawing` — для изображения графа (рисования)
- `add_edga` — добавление линий графа
- `shuffle` — перестановка вершин
- `get` — поиск по словарю

Алгоритм:

- 1) В отдельном файле указывается информация о графе: n — количество вершин, m — количество ребер и номера вершин и с какими они смежные.
- 2) Написаны функции для создания графа и изоморфного: добавления линии графа, отображение графа, перестановка вершин, путем их замены.
- 3) Создан словарь данного графа.
- 4) Перестановка вершин, для создания изоморфного графа и обновление словаря.
- 5) Доказывающий путем знания гамильтонова цикла, находит его в изоморфном графе.
- 6) Передача изоморфного графа проверяющему.
- 7) Проверяющий задает один из двух вопросов случайным образом вопрос с вероятностью 0,5: показать гамильтонов цикл изоморфного графа или доказать, что граф изоморфный (показать перестановку).

Результат работы программы

Изоморфный граф:



Результат при выборе первого вопроса:

```
Ответить на вопрос:
1 - Гамильтонов цикл изоморфного графа
2 - Доказать, что граф изоморфный
1
Гамильтонов цикл: ['3', '12', '4', '16', '9', '10', '2', '15', '6', '11', '8', '7', '1', '13', '5', '17', '20', '14', '19', '18']
```

Результат при выборе второго вопроса:

```
Ответить на вопрос:
1 - Гамильтонов цикл изоморфного графа
2 - Доказать, что граф изоморфный
2
Перестановка: {1: '3', 2: '12', 3: '4', 4: '16', 5: '9', 6: '10', 7: '2', 8: '15', 9: '6', 10: '11', 11: '8', 12: '7', 13: '1', 14: '13', 15: '5', 16: '17', 17: '20', 18: '14', 19: '19', 20: '18'}
Вершины и их пути: {'3': ['18', '12', '9'], '12': ['3', '14', '4'], '4': ['12', '17', '16'], '16': ['4', '9', '13'], '9': ['3', '16', '10'], '10': ['9', '2', '1'], '2': ['15', '18', '10'], '15': ['6', '2', '7'], '6': ['15', '11', '19'], '11': ['6', '20', '8'], '8': ['11', '5', '7'], '7': ['8', '1', '15'], '1': ['7', '13', '10'], '13': ['1', '16', '5'], '5': ['13', '17', '8'], '17': ['4', '20', '5'], '20': ['14', '17', '11'], '14': ['20', '12', '19'], '19': ['6', '14', '18'], '18': ['2', '3', '19']}
```

Листинг

main.py

```
import random
import networkx as nx
import matplotlib.pyplot as plt
import graph_params as gp

# Функция добавления линии графа
def add_edge(f_item, s_item, graph = None, color = 'black'):
    graph.add_edge(f_item, s_item, color=color)
    graph.add_edge(s_item, f_item, color=color)

# Рисует граф
def graph_drawing(graph_info, show_cycle = False, cycle = None):
    graph = nx.Graph()
    if not show_cycle and not cycle:
        graph.add_nodes_from(graph_info.keys())
    else:
        for node in graph_info.keys():
            if node == cycle[0] or node == cycle[len(cycle)-1]:
                graph.add_node(node, color = 'blue')
            else:
                graph.add_node(node, color = 'gray')
    # Создает связи между точками
    for vertex in graph_info.keys():
        for edge in graph_info.get(vertex):
            add_edge(vertex, edge, graph)
    # Раскрашивает связи если передан гамильтонов цикл
    if show_cycle and cycle:
        for i in range(0, 20):
            if i < 19:
                add_edge(cycle[i], cycle[i + 1], graph, color = 'g')
            else:
                add_edge(cycle[0], cycle[i], graph, color = 'g')
    # Рисует раскрашенный граф, если есть цикл
    if show_cycle:
        colors_edge = nx.get_edge_attributes(graph, 'color').values()
        colors_nodes = nx.get_node_attributes(graph, 'color').values()
        nx.draw_spring(graph, edge_color = colors_edge, node_color = colors_nodes, node_size =
1000, with_labels = True)
    # Рисует не раскрашенный граф
    else:
        nx.draw_spring(graph, edge_color = 'red', node_color = 'yellow', node_size = 1000,
with_labels = True)
    plt.show()

def shuffle(graph_info: dict, cycle: list):
    new_graph = {}
    new_cycle = []
    repl = [str(i) for i in graph_info.keys()]
    dictionary = {}
    # В ходе цикла меняет названия всех вершин графа
    for i in graph_info.keys():
        choise = random.choice(repl)
        repl.remove(choise)
        dictionary[i] = choise
```

```

# Переименовывает Вершины на полученные
for key in graph_info.keys():
    for value in graph_info.get(key):
        if new_graph.get(dictionary.get(key)):
            new_graph[dictionary.get(key)].append(dictionary.get(value))
        else:
            new_graph[dictionary.get(key)] = [dictionary.get(value)]
# Обновляет цикл в соответствии со словарем
for i in cycle:
    new_cycle.append(dictionary.get(i))
return new_graph, new_cycle, dictionary

if __name__ == "__main__":
    ig, gm, shuf = shuffle(gp.graph_info, gp.origin_graph_cycle)
    graph_drawing(ig)
    num = int(input("Ответить на вопрос:\n1 - Гамильтонов цикл изоморфного графа \n2 -
Доказать, что граф изоморфный\n"))
    if num == 1:
        print(f'Гамильтонов цикл: {gm}')
    if num == 2:
        print(f'Перестановка: {shuf}')
        print(f'Вершины и их пути: {ig}')

```