Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Дисциплина: Теория графов**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_К.Данил

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А. А. Крамаренко

**Тема работы: Проверка на существования графа**

**Ход работы:**

**Задача:**Написать программу проверки существования графа по заданным вершин.

**Решение задачи:**

1. Для написании программного кода был использован язык программирования Python, доп библиотеки не использовались.

2. Для существования графа были использованы следующие теоремы и леммы:

1. **Теорема:** Число нечетных вершин любого графа четно.

2. **Теорема:**Во всяком графе с *n* вершинами (*n* >2) всегда найдутся, по меньшей мере, две вершины с одинаковыми степенями.

3. **Теорема:** Если в графе с *n* вершинами (*n* >2) в точности две вершины имеют одинаковую степень, то в этом графе всегда найдется либо в точности одна вершина степени 0, либо в точности одна вершина степени *n* - 1.

4. **Лемма** о рукопожатиях

5. **Алгоритм** Хавел-Хакими для возможности построения простого графа

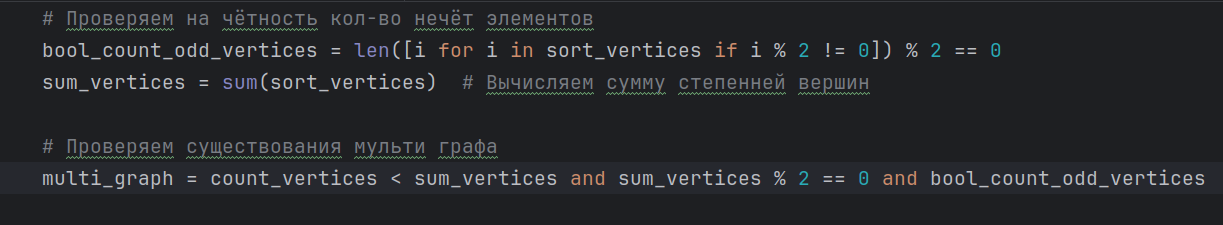
3. Графы которые были рассмотрены в ходе работы:

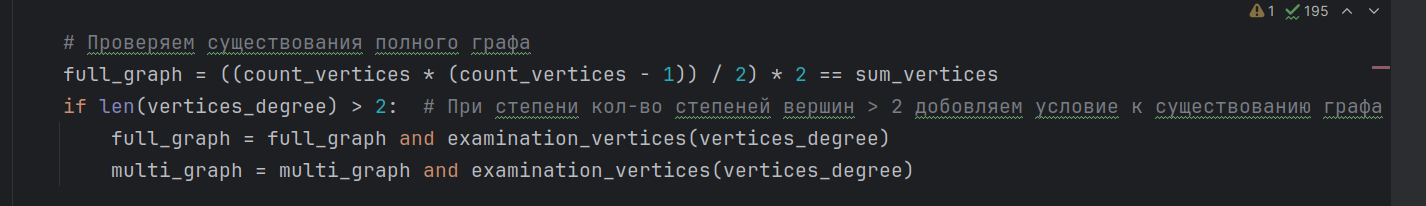
1. **«Мультиграф»**

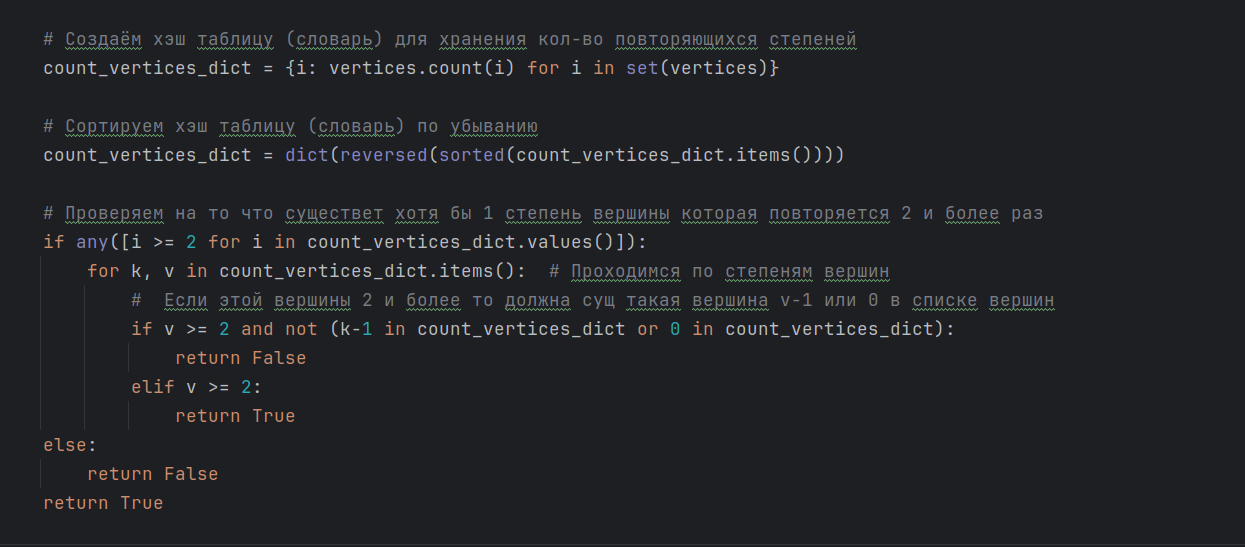
2. **«Полный граф»**

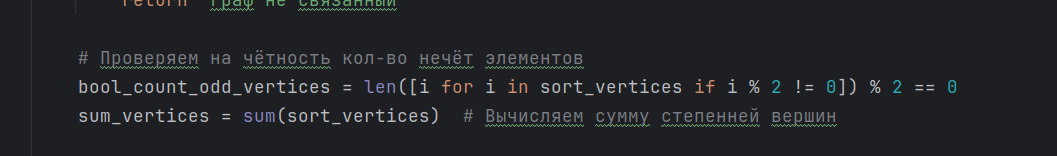
3. **«Простой граф»**

**Ход решения поставленной задачи:**

  **(Рисунок 1)**

 **(Рисунок 2)**

 **(Рисунок 3)**



**(Рисунок 4)**

Для существования любого графа была использована теорема **1**, а так же

осуществлялась проверка графа на кол-во степеней вершин, для проверки его существования по теореме **2** и **3**.

Теорема **1**,**2**,**3,4** рассмотрены на рисунке **1**,**2**,**3,4.**

Формула нахождения ребер полного графа рассмотрена на рисунке **1**.

Алгоритм Хавел-Хакими рассмотрен на рисунке.

**Проверка на существования «Мультиграфа»**

Если введённые степени вершин удовлетворяли теоремам **1,2,3,4**, то граф возможно построить. Такой граф будет называться **«Мультиграф»**.

**Проверка на существования «Полного графа»**

Для проверки существования полного графа, была использована теорема 1, а сумма ребер в таком графе считалась по формуле n \* (n — 1) / 2. Если выполнялось условие из теоремы **1,2,3**, то такой граф являлся **«Полным»**, а значит он существует.

**Проверка на существования «Простого графа»**

Для проверки существования простого графа, был использован алгоритм Хавел-Хакими. Данный алгоритм рекурсивно соединяет вершины по правилу:

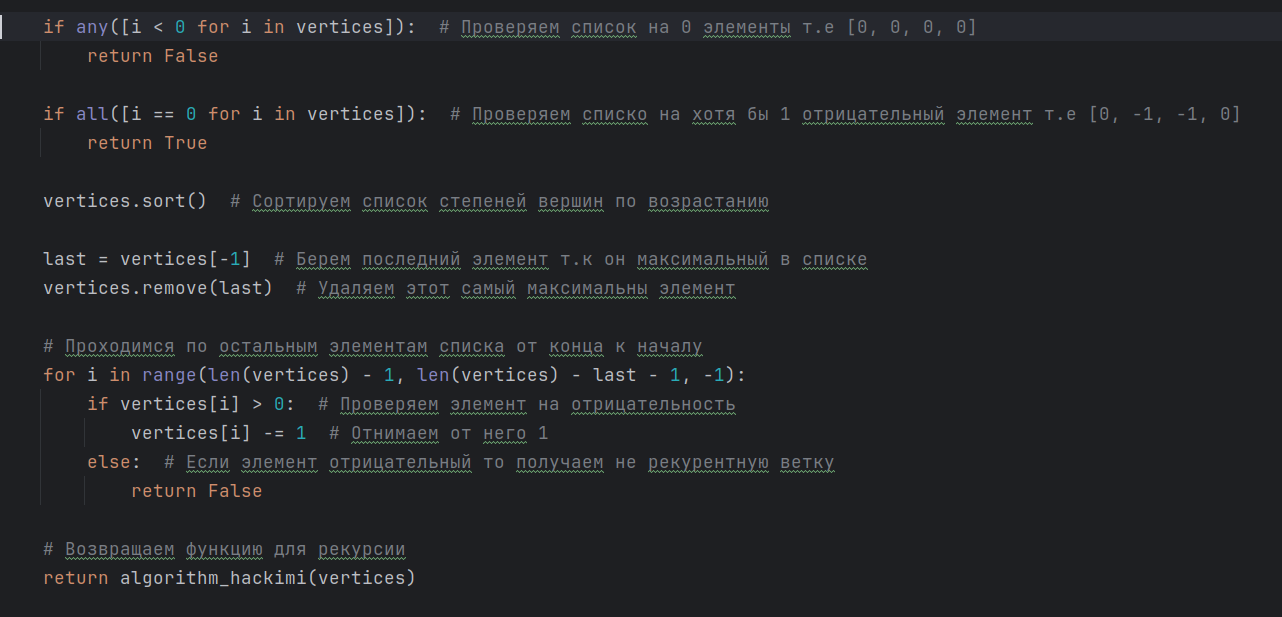
1. Сортируем список и проверяем нет ли там хотя бы одного отрицательного элемента, или все элементы списка равны 0. Если в списке существует отрицательный элемент, то простой граф построить нельзя, но если все элементы списка нулевые, то такой граф построить можно.

2. Удаляется последняя степени вершины в списке (она же и максимальная). Обозначим её как R.

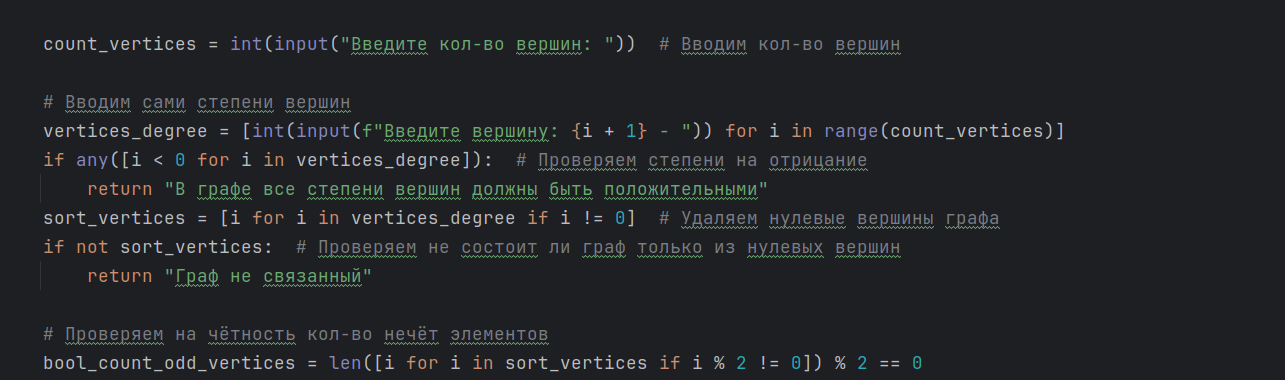
3. Уменьшаем R степеней вершин списка на 1.

4. Повторяем п.1, 2, 3 и так пока условие в п.1 не будет выполнено.

Если функция в котором выполняется данный алгоритм вернула False, то такой простой граф построить нельзя. В противном случае простой граф существует, и его можно построить. Так же мы не учитываем 0 степени вершин которые присутствуют в нашем списке.

 **(Рисунок 4)**

Ввод списка степеней вершин (Рисунок 5) проверяется на отрицательные элементы, а так же если список вершин при удалении нулей будет пуст, то данный граф будет являться не связанным.

 **(Рисунок 5)**

**Примеры обработки степеней вершин графа:**

