Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |
| --- | --- |
| Факультет (институт) | *Информационных технологий и компьютерных систем* |
|  |  |
| Кафедра | *Прикладная математика и фундаментальная информатика* |
|  |  |

**Расчетно-графическая работа**

|  |  |
| --- | --- |
| по дисциплине | ***Дискретная математика*** |
|  |  |
| на тему | Разработка программы «Вычисление связного остов графа с минимальным количеством ребер» |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 020-РГР-02.03.03-№ 11-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Мусияк Егор Алексеевич | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | *1* |  | Группа | | МО-221 | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.03*** | | |
|  | | | *Математическое обеспечение и администрирование информационных систем* | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | |  | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | | | |
|  |  |  |  | | | | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | дата, подпись руководителя |  |  |  |

Омск 2023

**Содержание**

|  |  |
| --- | --- |
| Текстовая формулировка задания………………………………………….. | 3 |
| Объяснение подхода к решению задания………………………………….. | 4 |
| Текст программы на языке Python................................................................... | 5 |
| Разработка пользовательского интерфейса………………………………... | 6 |
| Список литературы………………………………………………………….. | 8 |

**Текстовая формулировка задания**

В некотором городе есть метро, состоящее из N (1 <= N <= 1000) станций и M (0 <= M <= 500 000) линий, соединяющих их. Каждая линия обеспечивает проезд между какими-то двумя станциями в обе стороны. Между любой парой станций проведено не более одной линии. Сеть метро построена таким образом, чтобы с каждой станции можно было проехать на каждую (возможно, через промежуточные станции). Назовем это свойство связностью метро.

В связи с изобретением принципиально нового вида транспорта метро стало убыточным, и его работу решили оптимизировать. На заседании мэрии города было постановлено закрывать каждый год по одной линии, но так, чтобы связность метро каждый раз сохранялась.

По введенной информации о сети метро разработать порядок закрытия линий, до тех пор, пока связность остается.

**Формат входных данных**

Первая строка входного файла будет содержать числа N и M. В следующих М строках находится информация о линиях. Каждая из этих строк содержит через пробел числа Ai и Bi (Ai, Bi) — две станции, которые соединяет i-я линия.

**Формат выходных данных**

Выходной файл должен состоять из K строк. Каждая строка должна содержать два числа — номера двух станций, соединенных между собой. Вывести линии нужно в порядке их закрытия. Последнее значение количество лет, необходимых на закрытие линий.

**Объяснение подхода к решению задания**

После прочтения задачи я начал обдумывать варианты ее решения. Для начала можно представить данное метро как граф с ребрами (неориентированный граф), в котором станции – это вершины, а ветки метро, соединяющие их – ребра.

Также можно прийти к выводу, что для решения этой задачи нужно найти связный остов граф с минимально возможным числом ребер, для решения этой задачи я решил воспользоваться алгоритмом Прима.

Так как в алгоритме прима нужны веса ребер, то будем считать, что вес всех наших ребер одинаковый и равен 1. Далее выполняем алгоритм Прима пока не пройдемся по всем вершинам и не останется одна компонента связности, также не забываем записывать все взятые ребра в отдельный список. После алгоритма Прима создаем новый список, в который помещаем все ребра, которые содержатся в изначальном, но не содержатся в конечном списке, т.е. все те ребра, которые не влияют на связность графа.

На экран выводятся следующие данные: ребра, которые будут удалены в процессе перестройки (в формате: A B, где A – первая станция ребра, B - вторая) и количество лет, за которое будут удалены данные линии.

**Текст программы на языке Python**

import os

clear = lambda: os.system('cls')

clear()

s = 0

while (s!=3):

clear()

print("1. Решение задачи")

print("2. Об Авторе")

print("3. Выход...")

s = int(input())

if s == 1:

with open("C:/test.txt", "r" ) as f:

n, m =map(int, f.readline().split())

metro=[]

for i in range(m):

a, b = map(int, f.readline().split())

metro.append([a, b])

ribs = [metro[0]]

prima = [metro[0][0], metro[0][1]]

while len(prima)!=n:

for i in range(m):

if (metro[i][0] in prima) and (metro[i][1] in prima):

pass

else:

if (metro[i][0] in prima) and (not (metro[i][1] in prima) ):

prima.append(metro[i][1])

ribs.append(metro[i])

elif not( (metro[i][0] in prima) ) and (metro[i][1] in prima):

prima.append(metro[i][1])

ribs.append(metro[i])

print("Порядок закрытия ребер: ")

k=0

for i in metro:

if not(i in ribs):

print(i[0], i[1])

k+=1

print("k =", k)

elif s == 2:

print("Работу выполнил:")

print("Студент гр.МО-221 Мусияк Егор")

print("Контакты: mm.yegor@mail.ru")

print()

print("Для выхода нажмите любую кнопку...")

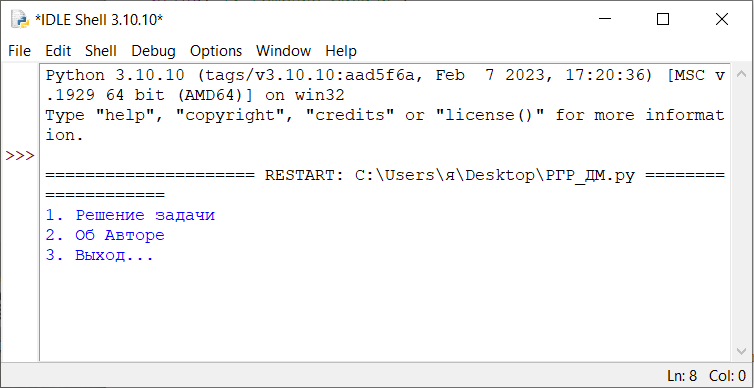
elif s == 3:

break

clear()

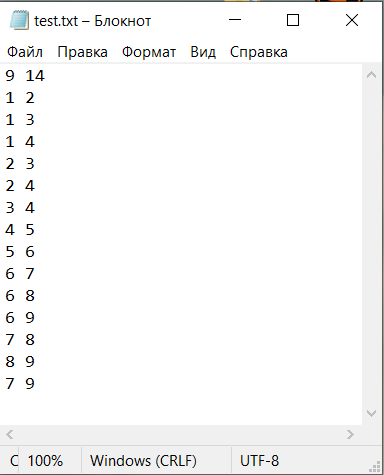
**Разработка пользовательского интерфейса**

На рисунке 1 представлено основное меню программы. Управление осуществляется нажатием клавиши с заданной цифрой.



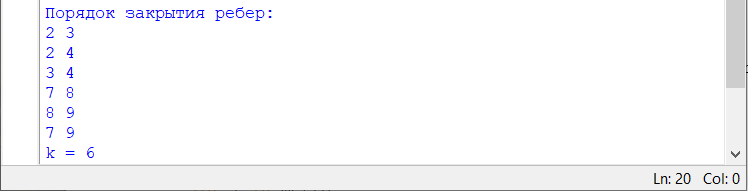
*Рисунок 1 – Основное меню программы*

На рисунке 2 представлены тестовые данные, которые были использованы для проверки правильности программы.



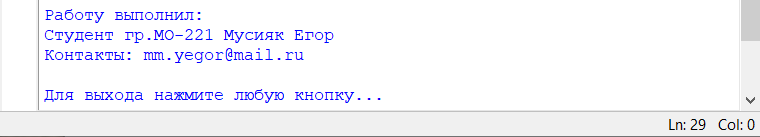
*Рисунок 2 – Тестовые данные*

На рисунке 3 представлен результат работы программы с данными с рисунка 2.



*Рисунок 3 – Ответ, полученный программой*

На рисунке 4 представлена информация об авторе программы, полученная нажатием клавиши «2».



*Рисунок 4 – Информация об авторе*

**Список литературы**

1. METANIT.COM [Электронный ресурс]. URL: <https://metanit.com> (Дата обращения: 21.04.2023).
2. Microsoft Lern [Электронный ресурс]. URL: <https://learn.microsoft.com/> (Дата обращения: 21.04.2023).
3. Python.org [Электронный ресурс]. URL: <https://www.python.org>

(Дата обращения: 22.04.2023).