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 |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 020-РГР-02.03.02-№ 9-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Есакова Кирилла Николаевича | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | *1* |  | Группа | | ФИТ-232 | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.02*** | | |
|  | | | *Фундаментальная информатика и информационные технологии* | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | | 26.05.2024 | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | | | |
|  |  |  |  | | | | | |  | |

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

Омск 2024

Содержание

[Введение 3](#_Toc167620494)

[Теоретическая часть 4](#_Toc167620495)

[Разработка кода 6](#_Toc167620496)

[Результаты программы 8](#_Toc167620497)

[Заключение 10](#_Toc167620498)

[Список используемой литературы 11](#_Toc167620499)

# **Введение**

В ходе работы необходимо написать программу, которая с помощью алгоритма Прима найдет минимальное остовное дерево. Описание задачи:

На графе изображена система связи предприятий (см. Рисунок1). Вероятность утечки информации при передаче (в процентах) соответствует весу рij ребра (xi, xj). Найти минимальную величину утечки информации при ее передаче между всеми предприятиями.

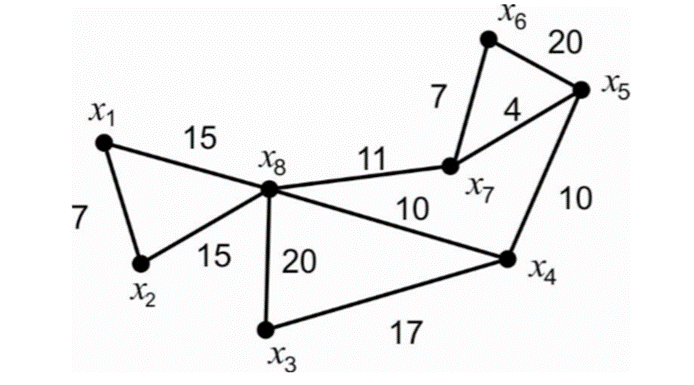


Рисунок 1 – Исходный граф

**Формат входных данных**

Во входном файле записано сначала число N (1 <= N <=100), определявшее количество точек передачи информации. Затем идет число M – количество соединений между точками передачи, далее идет описание потерь информации для каждой линии передачи. Каждое соединение между точками задается тремя числами – номерами точек передачи информации и величиной утечки информации.

**Формат выходных данных**

На экран выведите числа – минимальные значение утечки информации с указанием номеров пунктов или -1, если передача информации отсутствует.

# **Теоретическая часть**

Алгоритм Прима – алгоритм построения минимального остовного дерева взвешенного связного неориентированного графа. Алгоритм впервые был открыт в 1930 году чешским математиком Войцехом Ярником, позже переоткрыт Робертом Примом в 1957 году.

Описание алгоритма Прима:

На вход алгоритма подаётся связный неориентированный граф. Для каждого ребра задаётся его стоимость.

Сначала берётся произвольная вершина и находится ребро, инцидентное данной вершине и обладающее наименьшей стоимостью. Найденное ребро и соединяемые им две вершины образуют дерево. Затем, рассматриваются рёбра графа, один конец которых – уже принадлежащая дереву вершина, а другой – нет; из этих рёбер выбирается ребро наименьшей стоимости. Выбираемое на каждом шаге ребро присоединяется к дереву. Если в графе возник цикл, дуга игнорируется. Рост дерева происходит до тех пор, пока не будут исчерпаны все вершины исходного графа.

Результатом работы алгоритма является остовное дерево минимальной стоимости.

Рассмотрим работу алгоритма на примере нашей задачи.

Запишем множество вершин:

Выбираем произвольную вершину. Пусть это будет вершина

Выбираем инцидентную дугу с минимальным весом для данной вершины и соединяем новую. В данном случае соединяем вершину . Добавляем вершины в список помеченных. . Вес ребра , соединяющее эти вершины равен 7.

Таким образом повторяем до тех пор, пока число компонент связности не дойдет до 1.

Выбираем вершину , добавляем ее в список помеченных. . Вес ребра , соединяющее вершины равен 15.

Далее идем в вершину и добавляем ее в список помеченных. . Вес ребра , соединяющее вершины равен 10.

Выбираем вершину , добавляем ее в список помеченных. . Вес ребра , соединяющее вершины равен 17.

Затем необходимо пойти из вершины в , поскольку если мы пойдем из вершины в вершину , то получится цикл, что является недопустимым. Добавляем вершину в список помеченных. . Вес ребра , соединяющее вершины равен 10.

Далее идем в вершину и добавляем ее в список помеченных. . Вес ребра , соединяющее вершины равен 4.

Наконец идем в вершину и добавляем ее в список помеченных. . Вес ребра , соединяющее вершины равен 7. Число компонент связности равна 1, значит алгоритм завершил свою работу.

Считаем минимальное значение утечки информации между пунктами:

# **Разработка кода**

import heapq  
def prim(graph, starting\_vertex):  
 mst = []  
 visited = set([starting\_vertex])  
 edges = [  
 (cost, starting\_vertex, to)  
 for to, cost in graph[starting\_vertex].items()  
 ]  
 heapq.heapify(edges)  
  
 while edges:  
 cost, from\_vert, to\_vert = heapq.heappop(edges)  
 if to\_vert not in visited:  
 visited.add(to\_vert)  
 mst.append((from\_vert, to\_vert, cost))  
  
 for to\_next, cost in graph[to\_vert].items():  
 if to\_next not in visited:  
 heapq.heappush(edges, (cost, to\_vert, to\_next))  
  
 return mst  
  
def show\_student\_info():  
 print(“Есаков Кирилл Николаевич, ФИТ-232/2, 1 курс”)  
 input(“Нажмите Enter, чтобы вернуться в меню”)  
  
def show\_task\_description():  
 with open(‘RgrTask.txt’, ‘r’,encoding=”utf-8”) as file:  
 task = file.read()  
 print(task)  
 input(“Нажмите Enter, чтобы вернуться в меню”)  
  
def perform\_calculation(graph, starting\_vertex):  
 mst = prim(graph, starting\_vertex)  
 print(“\nПуть:”)  
 um = 0  
 for from\_vert, to\_vert, cost in mst:  
 print(f”{from\_vert} – {to\_vert} : {cost}”)  
 um += cost  
 print(f”\nМинимальное значение утечки информации:\n{um}\n”)  
 input(“Нажмите Enter, чтобы вернуться в меню”)  
  
def main():  
 filename = “input.txt”  
 with open(filename, “r”) as f:  
 num\_vertices = int(f.readline())  
 num\_edges = int(f.readline())  
  
 graph = {str(i): {} for i in range(1, num\_vertices + 1)}  
  
 for \_ in range(num\_edges):  
 start, end, weight = map(int, f.readline().split())  
 graph[str(start)][str(end)] = weight  
 graph[str(end)][str(start)] = weight  
  
 while True:  
 print(“Меню:”)  
 print(“1) Информация о студенте”)  
 print(“2) Описание задачи”)  
 print(“3) Вычисление”)  
 print(“4) Выход из программы”)  
  
 choice = input(“Выберите пункт меню: “)  
  
 if choice == ‘1’:  
 print(“-“\*100)  
 show\_student\_info()  
 print(“-“ \* 100)  
  
 elif choice == ‘2’:  
 print(“-“ \* 100)  
 show\_task\_description()  
 print(“-“ \* 100)  
  
 elif choice == ‘3’:  
 print(“-“ \* 100)  
 perform\_calculation(graph, “1”)  
 print(“-“ \* 100)  
  
 elif choice == ‘4’:  
 print(“Выход из программы”)  
 break  
  
 else:  
 print(“Некорректный выбор. Попробуйте снова.»)  
  
if \_\_name\_\_ == «\_\_main\_\_»:  
 main()

# **Результаты программы**

Рассмотрим результаты программы:

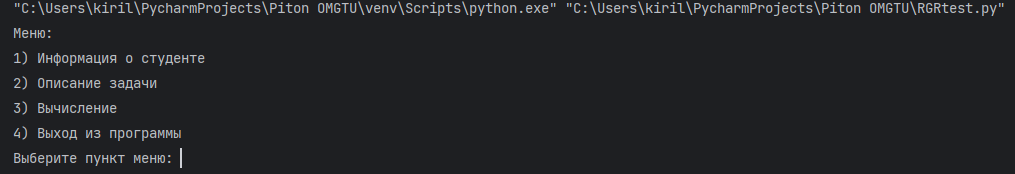


Рисунок 2 – Меню программы

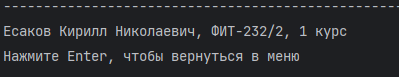


Рисунок 3 – Пункт «Информация о студенте»

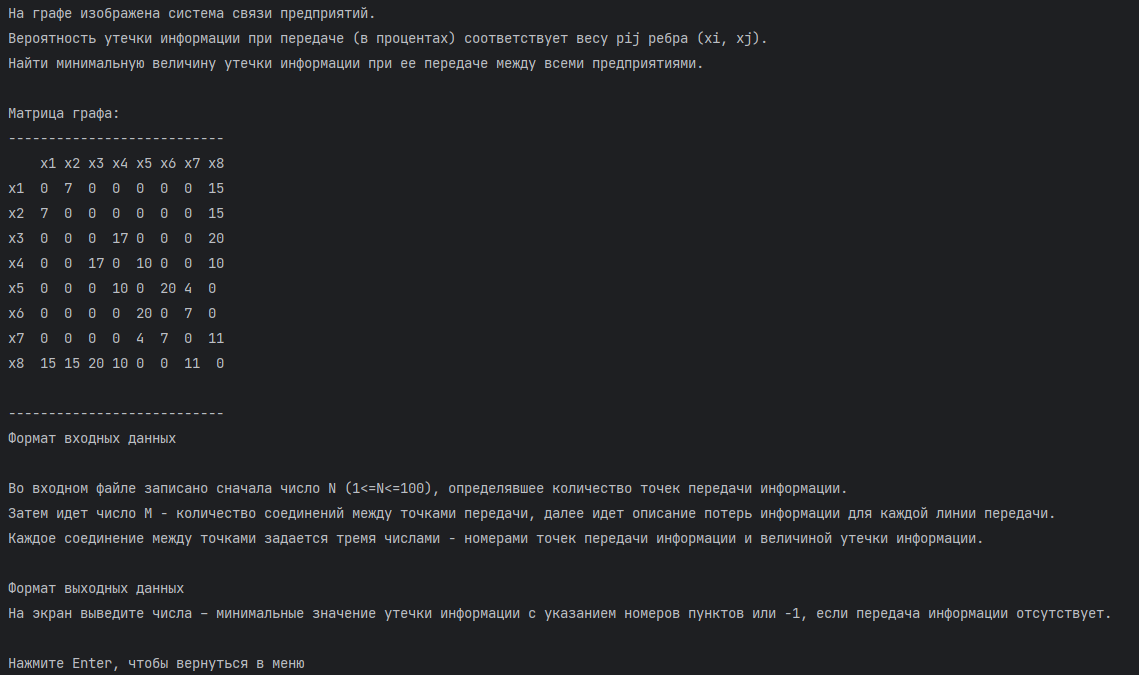


Рисунок 4 – Пункт «Описание задачи»

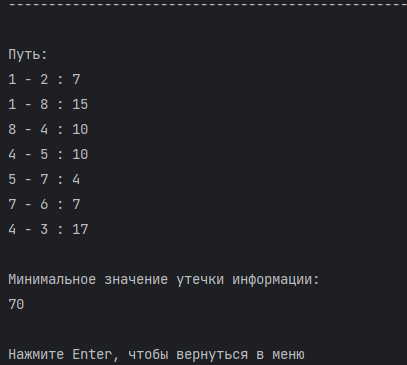


Рисунок 5 – Пункт «Вычисление»

# **Заключение**

В результате работы мы разработали программу, которая с помощью алгоритма Прима находит минимальное остовное дерево. Данная работа помогла закрепить знания, полученные во время изучения дисциплины «Дискретная математика».

# **Список используемой литературы**

1. Metanit, Руководство по языку программирования Python, <https://metanit.com/python/tutorial/> – (дата обращение: 25.05.2024)
2. Википедия, Алгоритм Прима, <https://ru.wikipedia.org/wiki/Алгоритм_Прима#Реализация> – (дата обращение: 25.05.2024)
3. *Применение алгоритмов на графах для решения прикладных задач*: метод. указания /Минобрнауки России, Омский государственный технический университет/; Е. А. Калиберда, И. В. Федотова – Омск: Изд-во ОмГТУ, 2021.