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-№ 13-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Мельникова Максима Кирилловича | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | *1* |  | Группа | | ФИТ-**231** | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.02*** | | |
|  | | | *Фундаментальная информатика и информационные технологии* | | | | | | | |
|  |  |  | 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 2024

**Содержание**

[Введение 3](#_Toc167034402)

[Постановка задачи 4](#_Toc167034403)

[Теоретическая часть 5](#_Toc167034404)

[Программная реализация 7](#_Toc167034405)

[Результаты 11](#_Toc167034406)

[Заключение 12](#_Toc167034407)

[Список использованной литературы 13](#_Toc167034408)

Введение

На сегодняшний день информационные технологии играют большую роль в жизни общества. Прогресс не стоит на месте, и теперь эти технологии применяются практически во всех сферах жизни человечества. Но для решения многих задач могут быть очень полезны знания дискретной математики.

Постановка задачи

Условие задачи (№13).

Андрей работает системным администратором и планирует создание новой сети в своей компании. Всего будет N хабов, они будут соединены друг с другом с помощью кабелей.

Поскольку каждый сотрудник компании должен иметь доступ ко всей сети, каждый хаб должен быть достижим от любого другого хаба – возможно, через несколько промежуточных хабов. Поскольку имеются кабели различных типов и короткие кабели дешевле, требуется сделать такой план сети (соединения хабов), чтобы максимальная длина одного кабеля была как можно меньшей. Есть еще одна проблема – не каждую пару хабов можно непосредственно соединять по причине проблем совместимости и геометрических ограничений здания. Андрей снабдит вас всей необходимой информацией о возможных соединениях хабов.

Необходимо помочь Андрею найти способ соединения хабов, который удовлетворит всем указанным выше условиям.

Теоретическая часть

Задача была решена с применением алгоритма Форда – Беллмана. Но для получения максимального количества набранных знаний в алгоритме использовались числа с противоположным знаком.

Рассмотрим пример с графом, заданным матрицей весов (рисунок 1), где вершины V являются комнатами, а дуги с весами – дверями с числами прибавляемых знаний.

Так как мы ищем максимальный путь от первой вершины до 4, то необходимо заменить числа в матрице на числа с противоположным знаком (рисунок 2).

Дальше строим новую таблицу (рисунок 3), где i – это номер итерации, а число за чертой – это вершина, из которой пришли. Так на итерации 0 путь до начальной вершины 1 равен 0.

На итерации 1 только в вершину 2 можно попасть из вершины 1, поэтому записываем соответствующее значение.

На итерации 2 из вершины 2 можно попасть в вершину 3 и вершину 4. Поэтому записываем соответствующие длины путей до этих вершин, которые являются суммами длины пути от вершины 1 до вершины 2 и длины пути от 2 до 3 и 4 соответственно.

На итерации 3 из вершины 3 можно попасть в вершину 2 и 4. Так как длина пути 1 – 2 меньше длины пути 1 – 2 – 3 – 2, то переходим к сравнению путей 1 – 2 – 4 и 1 – 2 – 3 – 4. Здесь уже длина пути 1 – 2 – 3 – 4 меньше и мы выбираем её.

Теперь для получения максимального пути необходимо изменить знаки у чисел на противоположные.

В результате ручного вычисления с помощью данного алгоритма мы получили максимальный путь до вершины 4, который равен 17.

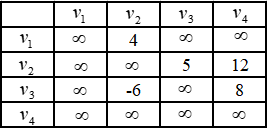


Рисунок 1 – Матрица весов

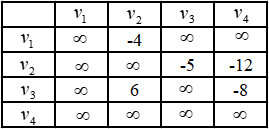


Рисунок 2 – Матрица весов с противоположными знаками

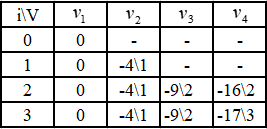


Рисунок 3 – Таблица работы алгоритма

Программная реализация

**Структура**

Код выполнен в среде PyCharm Community Edition на языке Python 3.12.

Программа состоит из трёх основных функций: Menu – функция, вызывающая меню, Alg – функция, выполняющая алгоритм, Create – функция, создающая выходной файл.

**Код программы**

import math

import os

num = 0

while num != 4:

print("1) Начать \n2) Условия задачи \n3) Об авторе \n4) Выход\nВыберите действие: ")

num = int(input())

if num == 1:

os.system('cls||clear')

print("Введите количество хабов в сети и количество количество возможных соединений хабов: ")

stroka = input().split(" ")

N, M = int(stroka[0]), int(stroka[1])

V = {1}

Spsk\_Result = []

Spsk\_Reber = [[math.inf, -1, -1]]

print("Введите все возможные соединения хабов: ")

for j in range(0, M):

srtoka = input().split(" ")

Spsk\_Reber.append([int(srtoka[0]), int(srtoka[1]), int(srtoka[2])])

def get\_min(Spsk\_Reber, V):

jm = [math.inf, -1, -1]

for j in V:

jj = min(Spsk\_Reber, key=lambda x: x[0] if (x[1] == j or x[2] == j) and (x[1] not in V or x[2] not in V) else math.inf)

if jm[0] > jj[0]:

jm = jj

return jm

while len(V) < N:

j = get\_min(Spsk\_Reber, V)

if j[0] == math.inf:

break

Spsk\_Result.append(j)

V.add(j[1])

V.add(j[2])

print(f"Всего использовано кабелей: {len(Spsk\_Result)}")

num = 0

for j in Spsk\_Result:

num += 1

print(f"Пара хабов №{num}: {j[1]}, {j[2]} \n Длина кабеля: {j[0]}]\n")

if num == 2:

os.system('cls||clear')

print(" Андрей работает системным администратором и планирует создание новой сети в своей компании. Всего будет N хабов, они будут соединены друг с другом с помощью кабелей.\n Поскольку каждый сотрудник компании должен иметь доступ ко всей сети, каждый хаб должен быть достижим от любого другого хаба – возможно, через несколько промежуточных хабов.\nПоскольку имеются кабели различных типов и короткие кабели дешевле, требуется сделать такой план сети (соединения хабов), чтобы максимальная длина одного кабеля была как можно меньшей.\nЕсть еще одна проблема – не каждую пару хабов можно непосредственно соединять по причине проблем совместимости и геометрических ограничений здания. Андрей снабдит вас всей необходимой информацией о возможных соединениях хабов.\n Необходимо помочь Андрею найти способ соединения хабов, который удовлетворит всем указанным выше условиям.\n")

if num == 3:

os.system('cls||clear')

print("Студент 1 курса ОмГТУ группы ФИТ-231, Мельников Максим Кириллович\n")

os.system('cls||clear')

Результаты

На основе выбранного пользователем пункта из отображаемого меню можно получить: результаты работы алгоритма (рисунок 4), условия задачи (рисунок 5), информацию об авторе программы (рисунок 6).

Результаты, полученные в результате выполнения программы и ручного вычисления на основе алгоритма Форда – Беллмана, полностью совпадают. Это означает исправную работу алгоритма программы.

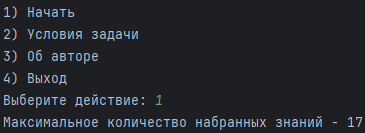


Рисунок 4 – Результаты выполнения программы

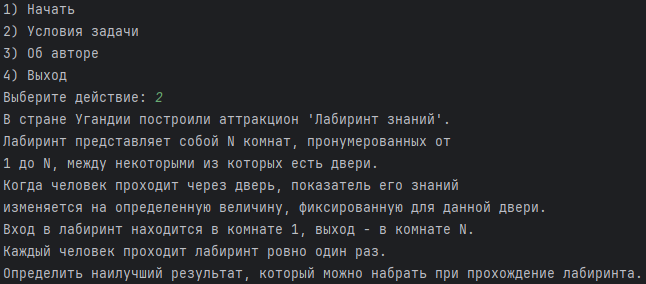


Рисунок 5 – Условия задачи

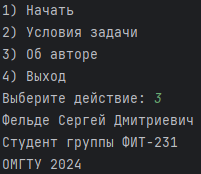


Рисунок 6 – Информация об авторе

Заключение

В ходе выполнения расчётно-графической работы была разработана и реализована программа для вычисления максимального пути на основе уже существующего алгоритма Форда – Беллмана.

Список использованной литературы

1. METANIT. https://metanit.com/python/ (Дата обращения: 19.04.24)
2. Python Lessons. https://pylessons.readthedocs.io/ru/latest/contents.html (дата обращения: 19.04.24)