

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

Дисциплина «Методы программирования»

Лабораторная работа №5

на тему «Графы и алгоритмы на графах – 2: минимальный каркас, эйлеров цикл и конденсация графа, алгоритм Форда-Фалкерсона»

Выполнил студент группы ВКБ31

Автайкин Алексей

г. Ростов-на-Дону

2021 г.

#### Задача Остовное дерево

Код программы:

class DSU:

    def \_\_init\_\_(self, k):

        self.parent = list(range(k))

        self.rank = [0] \* k  # ранк - это глубина дерева

    def make\_set(self):

        self.parent.append(len(self.parent))

        self.rank.append(0)

    def find\_set(self, v):

        if v == self.parent[v]:

            return v

        self.parent[v] = self.find\_set(self.parent[v])

        return self.parent[v]

    def union\_sets(self, a\_, b\_):

        a = self.find\_set(a\_)

        b = self.find\_set(b\_)

        if a != b:

            if self.rank[a] < self.rank[b]:

                a, b = b, a

            self.parent[b] = a

            if self.rank[a] == self.rank[b]:

                self.rank[a] += 1

if \_\_name\_\_ == '\_\_main\_\_':

    n, m = map(int, input().split())

    dsu = DSU(n)

    ribs = []

    for \_ in range(m):

        req = list(map(int, input().split()))

        req.reverse()

        ribs.append(req)

    ribs.sort()   # [[w, a, b], [...], ... ]

    ch = 0

    for i in ribs:

        if dsu.find\_set(i[1] - 1) != dsu.find\_set(i[2] - 1):

            dsu.union\_sets(i[1] - 1, i[2] - 1)

            #  print(dsu.parent)

            ch += i[0]

    print(ch)

Оценка программы:

