***Реализация алгоритма Форда-Фалкерсона***

import math

def get\_max\_vertex(k, V, S):

m = 0 # наименьшее допустимое значение

v = -1

for i, w in enumerate(V[k]):

if i in S:

continue

if w[2] == 1: # движение по стрелке

if m < w[0]:

m = w[0]

v = i

else: # движение против стрелки

if m < w[1]:

m = w[1]

v = i

return v

def get\_max\_flow(T):

w = [x[0] for x in T]

return min(\*w)

def updateV(V, T, f):

for t in T:

if t[1] == -1: # это исток

continue

sgn = V[t[2]][t[1]][2] # направление движения

# меняем веса в таблице для (i,j) и (j,i)

V[t[1]][t[2]][0] -= f \* sgn

V[t[1]][t[2]][1] += f \* sgn

V[t[2]][t[1]][0] -= f \* sgn

V[t[2]][t[1]][1] += f \* sgn

V = [[[0,0,1], [20,0,1], [30,0,1], [10,0,1], [0,0,1]],

[[20,0,-1], [0,0,1], [40,0,1], [0,0,1], [30,0,1]],

[[30,0,-1], [40,0,-1], [0,0,1], [10,0,1], [20,0,1]],

[[10,0,-1], [0,0,1], [10,0,-1], [0,0,1], [20,0,1]],

[[0,0,1], [30,0,-1], [20,0,-1], [20,0,-1], [0,0,1]],

]

N = len(V) # число вершин в графе

init = 0 # вершина истока (нумерация с нуля)

end = 4 # вершина стока

Tinit = (math.inf, -1, init) # первая метка маршруто (a, from, vertex)

f = [] # максимальные потоки найденных маршрутов

j = init

while j != -1:

k = init # стартовая вершина (нумерация с нуля)

T = [Tinit] # метки маршрута

S = {init} # множество просмотренных вершин

while k != end: # пока не дошли до стока

j = get\_max\_vertex(k, V, S) # выбираем вершину с наибольшей пропускной способностью

if j == -1: # если следующих вершин нет

if k == init: # и мы на истоке, то

break # завершаем поиск маршрутов

else: # иначе, переходим к предыдущей вершине

k = T.pop()[2]

continue

c = V[k][j][0] if V[k][j][2] == 1 else V[k][j][1] # определяем текущий поток

T.append((c, j, k)) # добавляем метку маршрута

S.add(j) # запоминаем вершину как просмотренную

if j == end: # если дошди до стока

f.append(get\_max\_flow(T)) # находим максимальную пропускную способность маршрута

updateV(V, T, f[-1]) # обновляем веса дуг

break

k = j

F = sum(f)

print(f"Максимальный поток равен: {F}")