Министерство образования Республики Беларусь Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Лабораторная работа №4

«Алгоритм Форда-Фалкерсона»

Выполнил: ст. гр. 953503 Басенко К.А.

Проверил: Дугинов О. И.

Минск 2022

# Постановка задачи

Задана сеть *G=(V, A)* с пропускной способностью *с*. Требуется найти максимальный поток в сети *G* из источника s в сток t.

# Описание алгоритма метода

Вход: *G(V, A), c, s, t*

Выход: максимальный поток *f* в *G из s в t*

1. Для каждой дуги исходного графа строим обратную ей дугу с пропускной способностью *с = 0.*
2. Строим нулевой поток *f* для дуг графа *G.*
3. Строим вспомогательную сеть *Gf=G(V, A), cf*, пропускные способности дуг, которой вычисляются по формуле

𝑐 (𝑎) = 𝑐(𝑎) − 𝑓(𝑎) + 𝑓(𝑎)*,*

𝑓

где *с(a)* - пропускная способность соответствующей дуги из графа G, *f(a)* - поток по этой дуге, 𝑓(𝑎) - поток по обратной дуге.

1. Из *Gf* удаляем все дуги с нулевой пропускной

способностью. В полученной сети ищем *(s, t)*-путь

1. Если такого пути нет, то STOP. Текущий поток *f* - это максимальный поток в *G*.
2. Находим в *Gf (s, t)*-путь *P*.
3. Находим θ = 𝑐 (𝑎) .

𝑓

1. Вернем в сеть Gf все дуги с нулевыми пропускными способностями, которые были убраны на шаге 4.
2. Строим дополнительный поток *fp* сети *Gf* вдоль пути *P*.

𝑓 : 𝐴 → 𝑅 ∀ 𝑎 ∈ 𝐴 𝑓 (𝑎) = {0, если 𝑎 ∉ 𝐴(𝑃) θ, если 𝑎 ∈ 𝐴(𝑃)

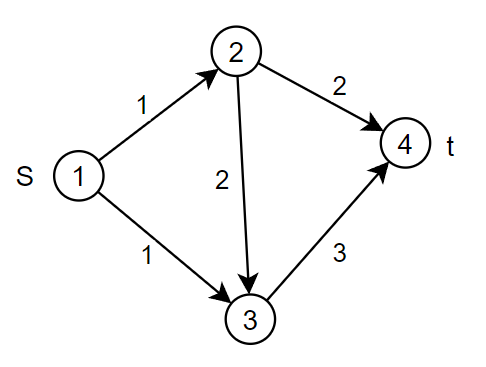
𝑝 𝑝

1. 𝑓 ← 𝑓 ⨁𝑓 *.* Переходим на шаг 2.

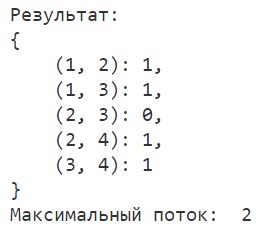
𝑝

# Результат работы

# Тест 1

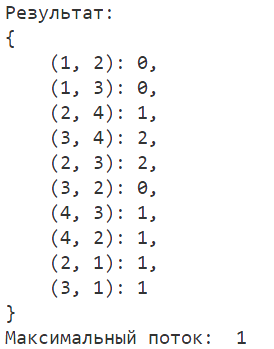


Результат:



# Тест 2

# 



**Код программы**

from math import inf

def ford\_falkerson(V, a, s, t):

ra = {(i[1], i[0]): 0 for i in a}

f = a.copy() | ra.copy()

for i in f:

f[i] = 0

sup\_arcs = a | ra

for i in a:

sup\_arcs[i] = a[i] - f[i] + f[(i[1], i[0])]

sup\_arcs[(i[1], i[0])] = ra[(i[1], i[0])] - f[(i[1], i[0])] + f[i]

while True:

labels = {i: None for i in V}

labels[s] = 0

q = [s]

while len(q) != 0:

point = q.pop(0)

out\_arcs = {i: sup\_arcs[i]

for i in sup\_arcs

if sup\_arcs[i] != 0 and i[0] == point}

for i in out\_arcs:

if labels[i[1]] == None:

labels[i[1]] = i

q.append(i[1])

if t in q:

break

if len(q) == 0:

# STOP

f = {i: f[i] for i in f if i in a}

max\_flow = sum(f[i] for i in f if i[0] == s)

return f, max\_flow

path = []

path.append(t)

while path[len(path) - 1] != s:

point = path[len(path) - 1]

path.append(labels[point][0])

path.reverse()

teta = inf

for i in range(1, len(path)):

if sup\_arcs[(path[i - 1], path[i])] < teta:

teta = sup\_arcs[(path[i - 1], path[i])]

for i in range(1, len(path)):

f[(path[i - 1], path[i])] += teta

sup\_arcs[(path[i - 1], path[i])] -= teta

sup\_arcs[(path[i], path[i - 1])] += teta

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

V = [

1, 2, 3, 4

]

a = {

(1, 2): 1,

(1, 3): 1,

(2, 3): 2,

(2, 4): 2,

(3, 4): 3,

}

s = 1

t = 4

f, max\_flow = ford\_falkerson(V, a, s, t)

print("Результат:", f)

print("Максимальный поток: ", max\_flow)