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 *c*. 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 *c = 0.*

2. Строим нулевой поток *f* для дуг графа *G.*

3. Строим вспомогательную сеть G_f=*G(V, A)*, c_f, пропускные способности дуг, которой вычисляются по формуле

$$c_f(a) = c(a) - f(a) + f(\overline{a}),$$

где *c(a)* - пропускная способность соответствующей дуги из графа G, *f(a)* - поток по этой дуге, $f(\overline{a})$ - поток по обратной дуге.

4. Из G_f удаляем все дуги с нулевой пропускной способностью. В полученной сети ищем *(s, t)*-путь

5. Если такого пути нет, то STOP. Текущий поток *f* - это максимальный поток в *G*.

6. Находим в G_f *(s, t)*-путь *P*.

7. Находим $\theta = c_f(a)$.

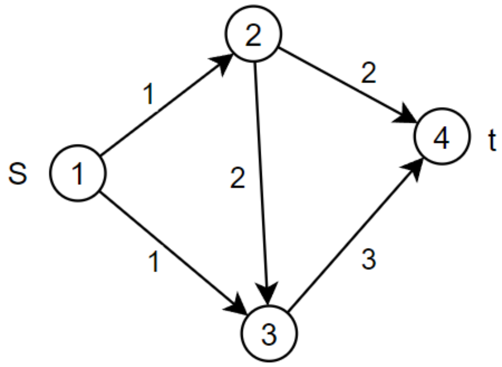
8. Вернем в сеть G_f все дуги с нулевыми пропускными способностями, которые были убраны на шаге 4.

9. Строим дополнительный поток f_p сети G_f вдоль пути *P*.

$$f_p : \overline{A} \to R \quad \forall a \in \overline{A} \ f_p(a) = \{0, \text{ если } a \notin A(P) \ \theta, \text{ если } a \in A(P)$$

10. $f \leftarrow f \oplus f_p$. Переходим на шаг 2.

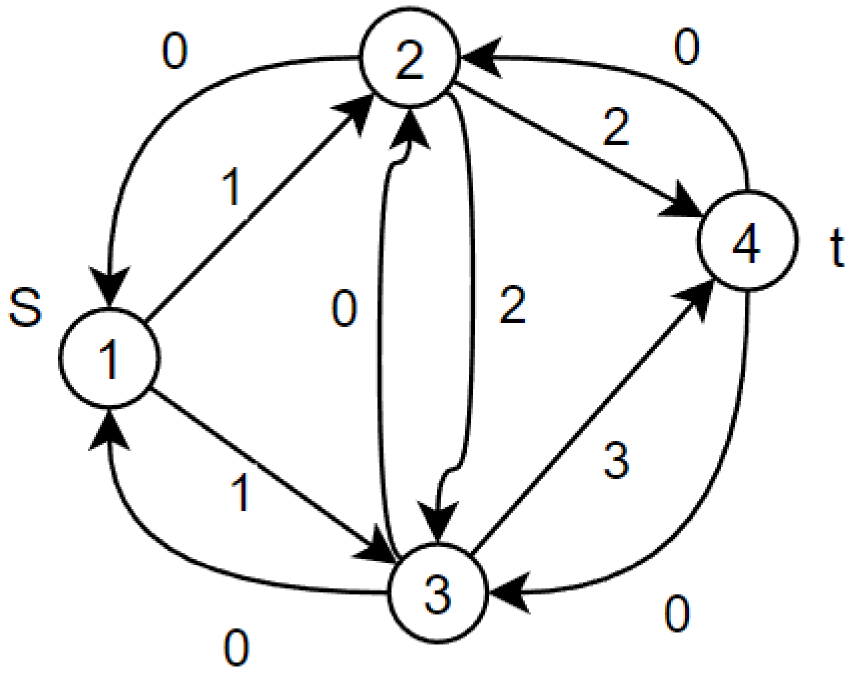
Результат работы
Тест 1



Результат:

```
Результат:
{
    (1, 2): 1,
    (1, 3): 1,
    (2, 3): 0,
    (2, 4): 1,
    (3, 4): 1
}
Максимальный поток:  2
```

Тест 2




```
Результат:
{
    (1, 2): 0,
    (1, 3): 0,
    (2, 4): 1,
    (3, 4): 2,
    (2, 3): 2,
    (3, 2): 0,
    (4, 3): 1,
    (4, 2): 1,
    (2, 1): 1,
    (3, 1): 1
}
Максимальный поток:  1
```

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__':
```

```python
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)
```