

# Алгоритмы на графах

Модуль 3. Потоки в сетях и паросочетания.

Лекция 13.

Нахождение максимального  
потока.

Адигеев Михаил Георгиевич

2023

# План лекции

1. Леммы о свойствах потоков и разрезов.
2. Увеличивающая цепь.
3. Критерий оптимальности потока в терминах увеличивающей цепи.
4. Теорема и алгоритм Форда-Фалкерсона.

# Определения

Пусть  $S \subseteq V$  – подмножество вершин, содержащее источник  $s$ .

Обозначим  $T = V \setminus S$  – подмножество, содержащее сток  $t$ .

*Разрез*  $(S, T)$  – множество дуг  $e = (u, v): u \in S, v \in T$ .

Аналогично:  $(T, S)$  – множество дуг  $e = (u, v): u \in T, v \in S$ .

Обозначим:  $f(X, Y) = \sum_{e \in (X, Y)} f(e)$ .

# Леммы

**Лемма 1.** Для любого потока  $f$  и любого разреза  $(S, T)$  справедливо:  $\hat{f} = f(S, T) - f(T, S)$ .

Доказательство.

Рассмотрим величину

$$\Delta = \sum_{v \in T} \left( \sum_{e \in \alpha(v)} f(e) - \sum_{e \in \beta(v)} f(e) \right)$$

Эту величину можно рассчитать двумя способами.

# Леммы

$$\Delta = \sum_{v \in T} \left( \sum_{e \in \alpha(v)} f(e) - \sum_{e \in \beta(v)} f(e) \right)$$

1)  $\Delta = \hat{f}$ .

Это следует из того, что величина в скобках  $\sum_{e \in \alpha(v)} f(e) - \sum_{e \in \beta(v)} f(e)$  в силу условия баланса равна нулю для всех вершин кроме  $t$ . А для  $t$  она равна величине потока.

2) Рассмотрим, как входит в  $\Delta$  величина  $f(e)$  для разных дуг  $e = (u, v)$ .

Если  $u, v \in T$ , то  $f(e)$  входит в  $\Delta$  один раз с плюсом и один раз с минусом. Поэтому такие дуги вносят лишь 0 в итоговую величину  $\Delta$ .

# Леммы

$$\Delta = \sum_{v \in T} \left( \sum_{e \in \alpha(v)} f(e) - \sum_{e \in \beta(v)} f(e) \right)$$

Если  $u, v \in S$ , то  $f(e)$  вообще не входит в  $\Delta$ , поэтому такие дуги тоже вносят 0 в итоговую величину  $\Delta$ .

Если  $u \in S, v \in T$ , то  $f(e)$  входит в  $\Delta$  с плюсом. Поэтому сумма  $f(e)$  по таким дугам даёт  $f(S, T)$ .

Если  $u \in T, v \in S$ , то  $f(e)$  входит в  $\Delta$  с минусом. Поэтому сумма  $f(e)$  по таким дугам даёт  $-f(T, S)$ .

Итого:  $\Delta = f(S, T) - f(T, S)$ .

# Леммы

Введём обозначение:  $c(S, T) = \sum_{e \in (S, T)} c(e)$ .

**Лемма 2.** Для любого потока  $f$  и любого разреза  $(S, T)$  справедливо:  $\hat{f} \leq c(S, T)$ .

Доказательство.

По лемме 1 справедливо  $\hat{f} = f(S, T) - f(T, S)$ .

Из определения потока следует, что  $0 \leq f(e) \leq c(e)$ .

Поэтому:  $\hat{f} = f(S, T) - f(T, S) \leq c(S, T) - 0 = c(S, T)$ .

# Леммы

**Лемма 3.** Если для потока  $f$  и разреза  $(S, T)$  выполняется равенство  $\hat{f} = c(S, T)$ , то поток  $f$  является максимальным, а разрез  $(S, T)$  является минимальным.

Доказательство.

Непосредственно следует из леммы 2.

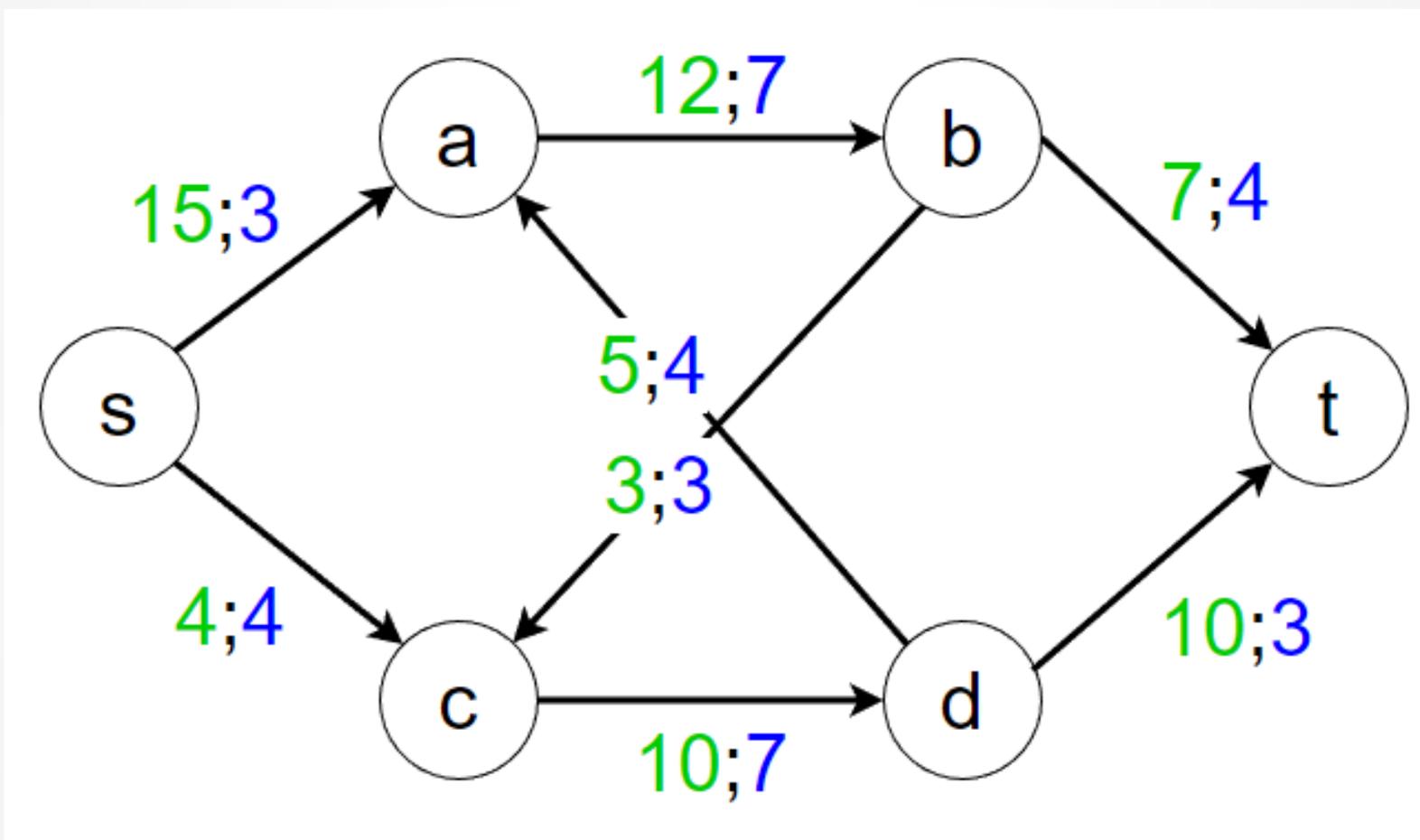
# Леммы

Пусть на графе  $G(V, E)$  с заданной пропускной способностью  $c: E \rightarrow R_+$  и потоком  $f: E \rightarrow R_+$ . Введём ещё одно понятие.

**Увеличивающей цепью  $\pi$**  будем называть цепь, соединяющую вершины  $s$  и  $t$ , и удовлетворяющую условиям:

- Если цепь  $\pi$  проходит по дуге  $e$  в прямом направлении, то  $f(e) < c(e)$ .
- Если цепь  $\pi$  проходит по дуге  $e$  в обратном направлении, то  $f(e) > 0$ .

# Леммы



# Леммы

**Лемма 4.** Поток  $f$  является максимальным  $\Leftrightarrow$  на графе нет увеличивающей цепи.

Доказательство.

$\Rightarrow$  Пусть поток  $f$  является максимальным. И предположим, что для  $f$  существует увеличивающая цепь  $\pi$ . Пройдём по этой цепи и вычислим величину  $\delta = \min_{e \in \pi} \Delta(e)$ , где

$$\Delta(e) = \begin{cases} c(e) - f(e) > 0, & \text{если } \pi \text{ проходит по ориентации } e, \\ f(e) > 0, & \text{если } \pi \text{ проходит против ориентации } e. \end{cases}$$

Очевидно, что  $\delta > 0$ .

# Леммы

Построим новый поток  $f'$  по следующему правилу:

$$f'(e) = \begin{cases} f(e) + \delta, & \text{если } \pi \text{ проходит по ориентации } e, \\ f(e) - \delta, & \text{если } \pi \text{ проходит против ориентации } e, \\ f(e), & \text{если } \pi \text{ не проходит через } e. \end{cases}$$

По построению,  $f'$  удовлетворяет условиям потока. При этом,

$\widehat{f}' = \widehat{f} + \delta$ . Отсюда следует, что поток  $f$  не максимален.

Получили противоречие.

# Леммы

$\Leftarrow$ . Предположим, что для потока  $f$  на графе нет увеличивающей цепи (из  $s$  в  $t$ ).

Определим множество  $S \subset V$  как множество вершин, для которых существует увеличивающая цепь из  $s$ , и обозначим  $T = V \setminus S$ . Очевидно, что  $s \in S, t \in T$ .

Тогда для любой дуги  $e = (u, v)$ , у которой концы лежат в разных подмножествах, справедливо:

- если  $e \in (S, T)$ , то дуга **насыщена**, т.е.  $f(e) = c(e)$ ,
- если  $e \in (T, S)$ , то дуга **не нагружена**, т.е.  $f(e) = 0$

(иначе оба конца дуги принадлежали бы  $S$ ).

# Леммы

Отсюда следует, что:  $f(S, T) = c(S, T)$ ,  $f(T, S) = 0$ .

Поэтому  $c(S, T) = \hat{f}$ , и по лемме 3 поток  $f$  максимален.

Лемма доказана.

# Алгоритм Форда-Фалкерсона

**Теорема (Форда-Фалкерсона)**. Максимальный поток равен пропускной способности минимального разреза:  $\max_f \hat{f} = \min_{(S,T)} c(S, T)$ .

Доказательство.

В силу леммы 3, для доказательства теоремы достаточно показать, что можно построить поток  $f$  и разрез  $(S, T)$ , такие, что  $\hat{f} = c(S, T)$ .

Рассмотрим алгоритм Форда-Фалкерсона, который строит поток с помощью процедуры, описанной в доказательстве леммы 4.

# Алгоритм Форда-Фалкерсона

1. Инициализация: присвоить потоку  $f$  допустимое значение, например:  $f(e) \equiv 0$ .

2.  $S := \{s\}$ . // Множество помеченных вершин.

$\delta := +\infty$ ;

Для всех  $v \in V$  выполнить:  $p[v] := NULL$ .

3. Пока  $t \notin S$  и есть дуга  $e = (u, v)$ , удовлетворяющая одному из условий:

a)  $u \in S, v \notin S, f(e) < c(e)$ .

b)  $u \notin S, v \in S, f(e) > 0$ .

Добавить в  $S$  ещё не помеченный конец дуги  $e$ :

a)  $S := S \cup \{v\}, p[v] := u, \delta := \min\{\delta, c(e) - f(e)\}$ ;

b)  $S := S \cup \{u\}, p[u] := v, \delta := \min\{\delta, f(e)\}$ ;

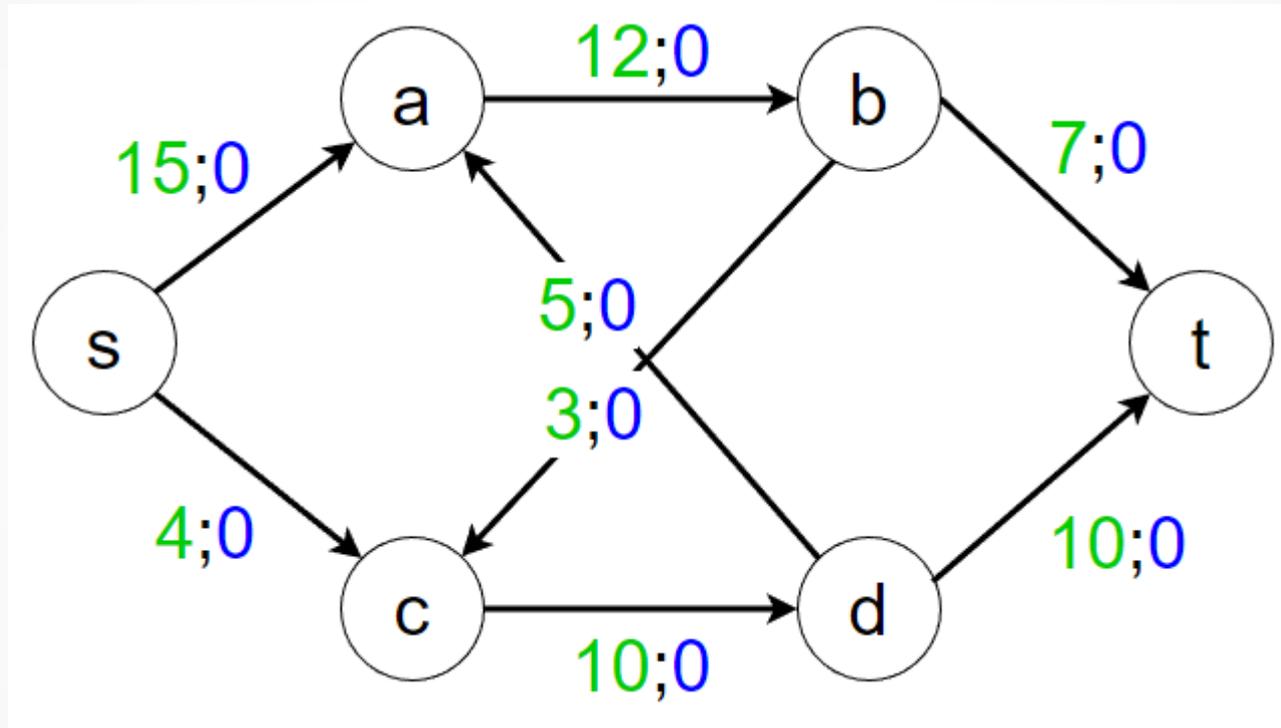
# Алгоритм Форда-Фалкерсона

4. Если  $t \notin S$ , то для текущего потока  $f$  не существует увеличивающей цепи. Следовательно,  $f$  является максимальным потоком. Остановить алгоритм и вернуть  $f$ .

Если  $t \in S$ , то по указателям  $p[]$  мы можем восстановить увеличивающую цепь и с её помощью построить (см. доказательство леммы 4) новый поток величины на  $\delta$  больше текущего. В этом случае возвращаемся к шагу 2.

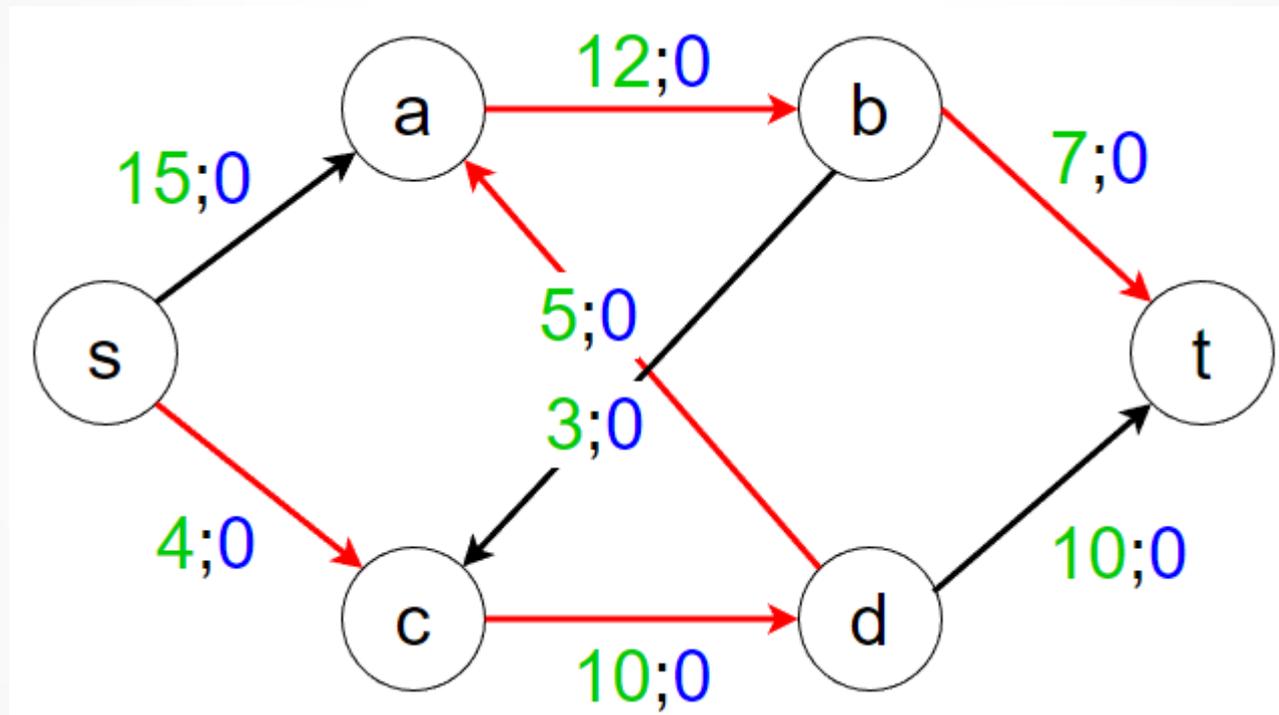
Теорема доказана.

# Алгоритм Форда-Фалкерсона



Строим увеличивающую цепь:  $s \rightarrow c \rightarrow d \rightarrow a \rightarrow b \rightarrow t$ .

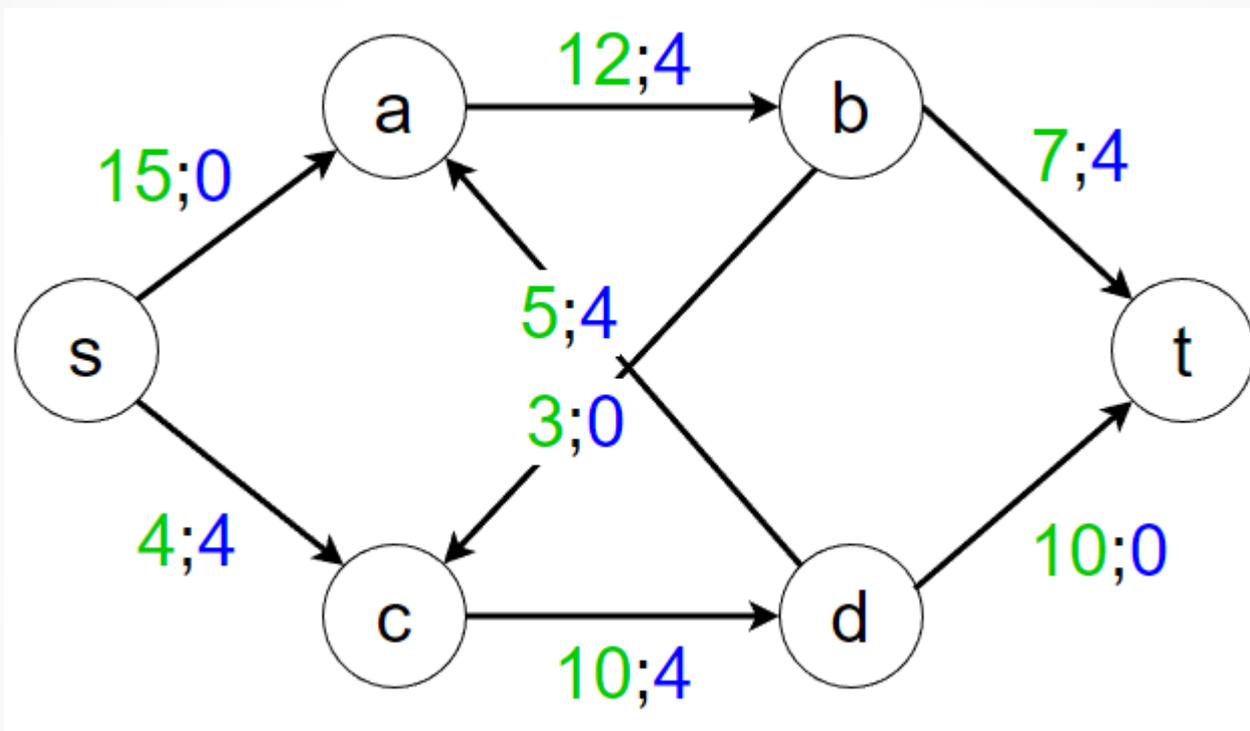
# Алгоритм Форда-Фалкерсона



$$\delta = \min\{4, 10, 5, 12, 7\} = 4$$

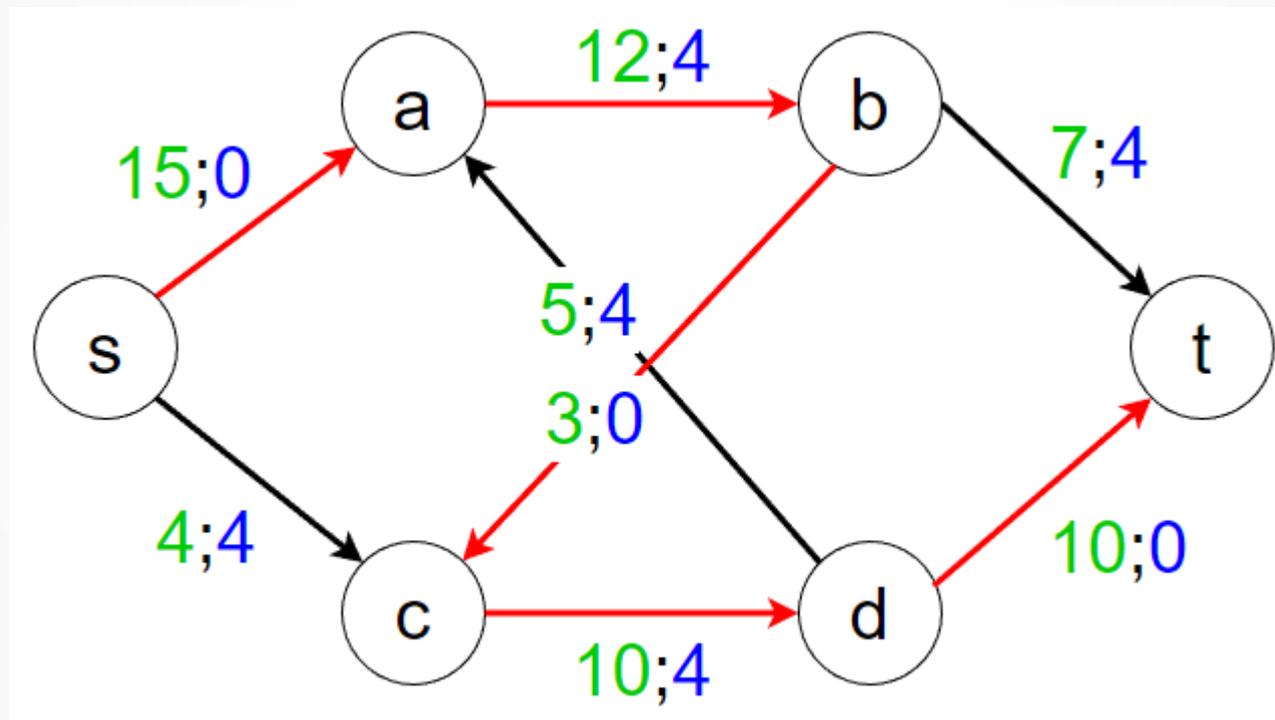
Увеличиваем поток:

# Алгоритм Форда-Фалкерсона



Строим увеличивающую цепь:  $s \rightarrow a \rightarrow b \rightarrow c \rightarrow d \rightarrow t$ .

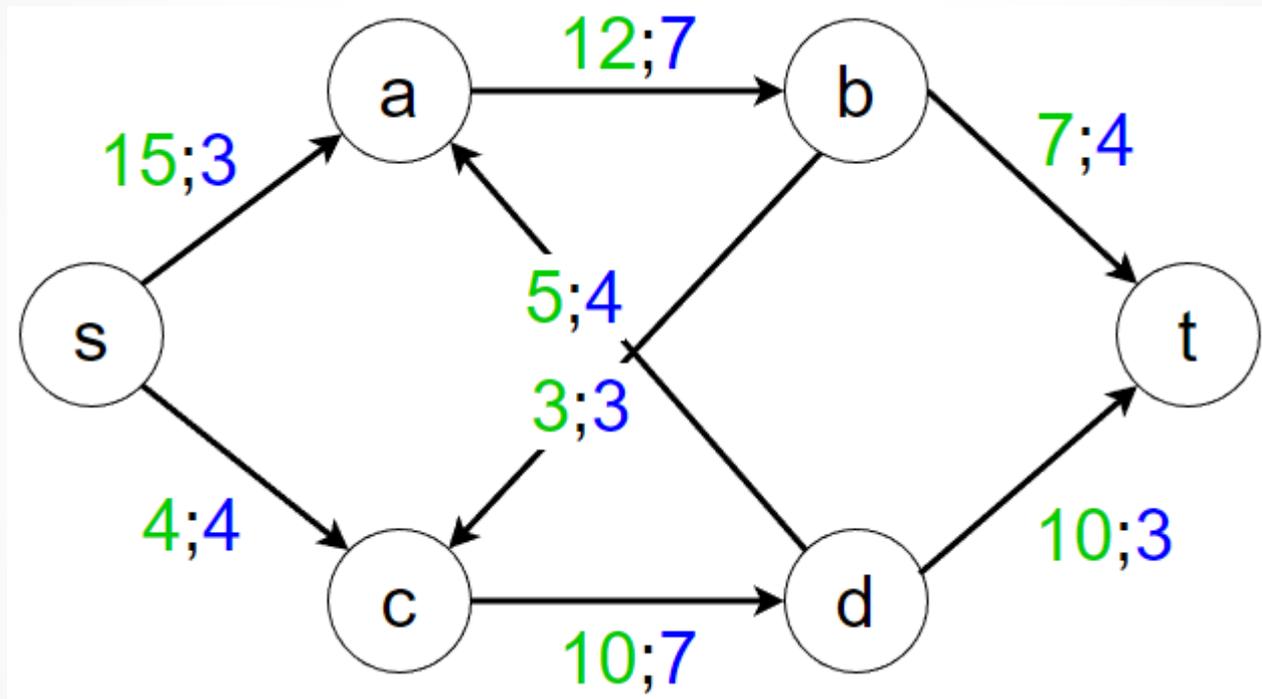
# Алгоритм Форда-Фалкерсона



$$\delta = \min\{15, 8, 3, 6, 10\} = 3$$

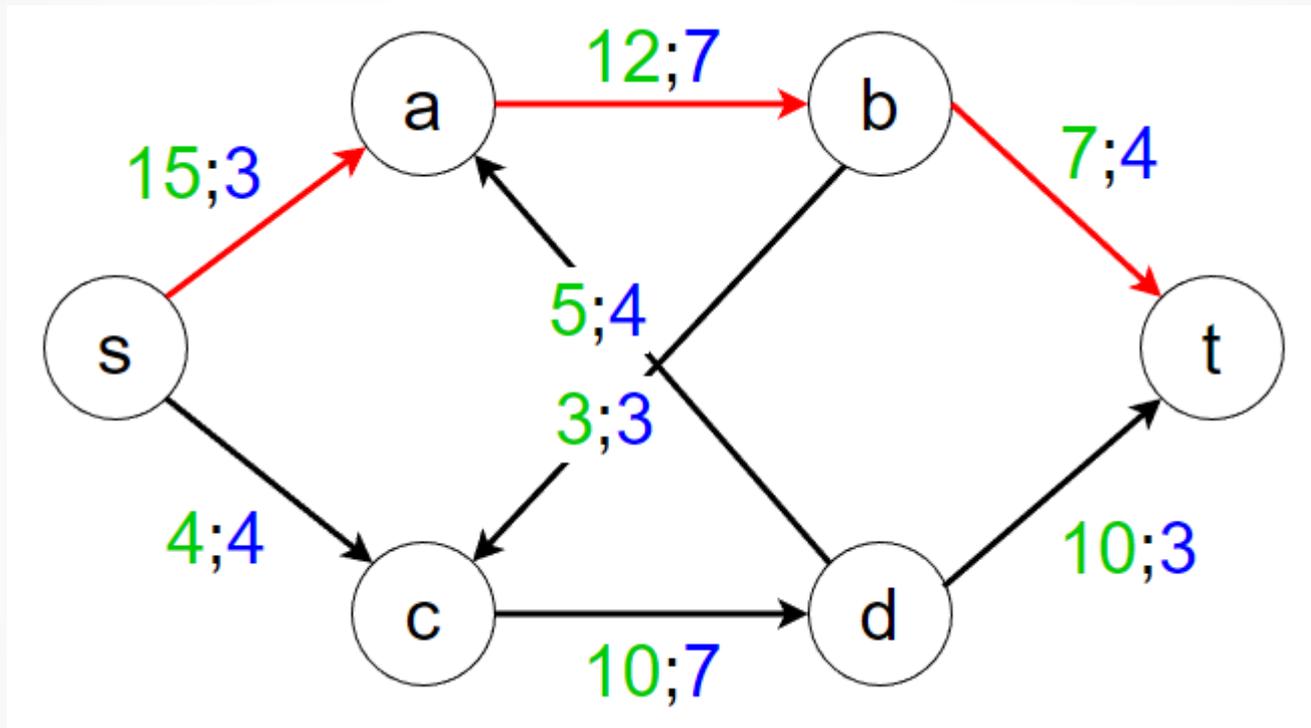
Увеличиваем поток:

# Алгоритм Форда-Фалкерсона



Строим увеличивающую цепь:  $s \rightarrow a \rightarrow b \rightarrow t$ .

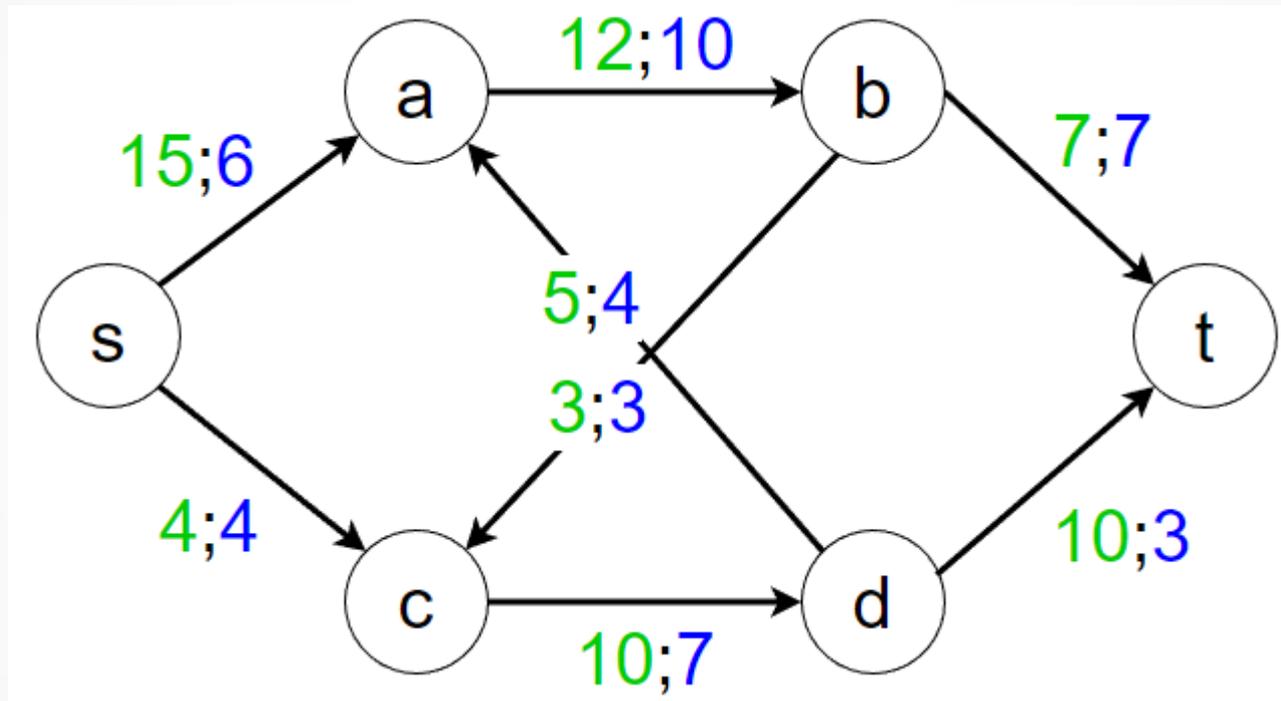
# Алгоритм Форда-Фалкерсона



$$\delta = \min\{12, 5, 3\} = 3$$

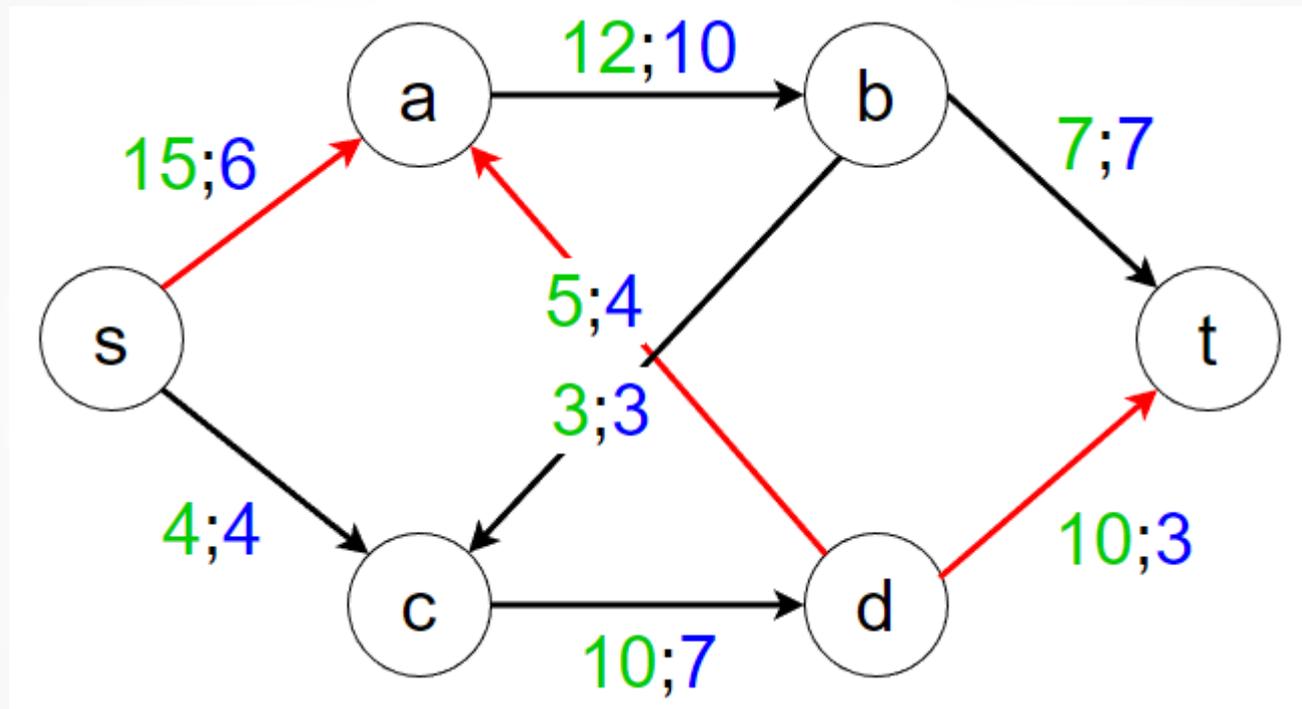
Увеличиваем поток:

# Алгоритм Форда-Фалкерсона



Строим увеличивающую цепь:  $s \rightarrow a \rightarrow d \rightarrow t$ .

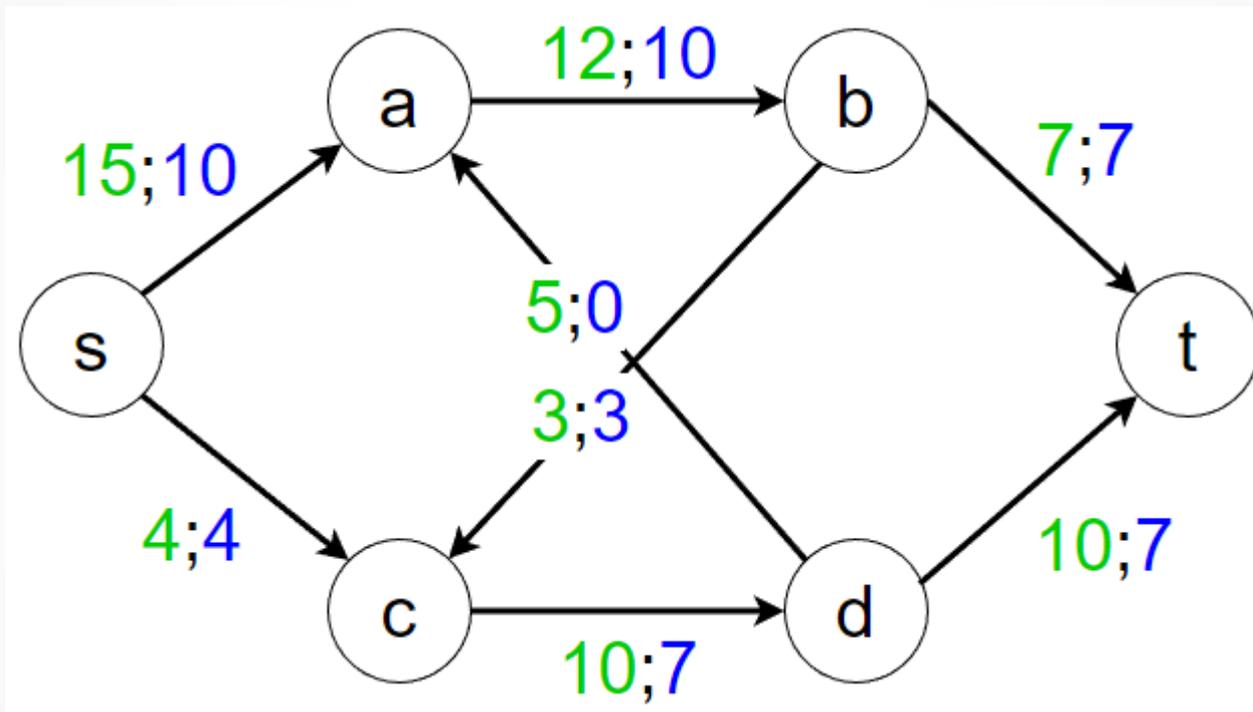
# Алгоритм Форда-Фалкерсона



$$\delta = \min\{9, 4, 7\} = 4$$

Увеличиваем поток:

# Алгоритм Форда-Фалкерсона



Увеличивающей цепи нет.  $S = \{s, a, b\}$ .

Текущий поток оптимален.  $\hat{f} = 14$ .

# Алгоритм Форда-Фалкерсона

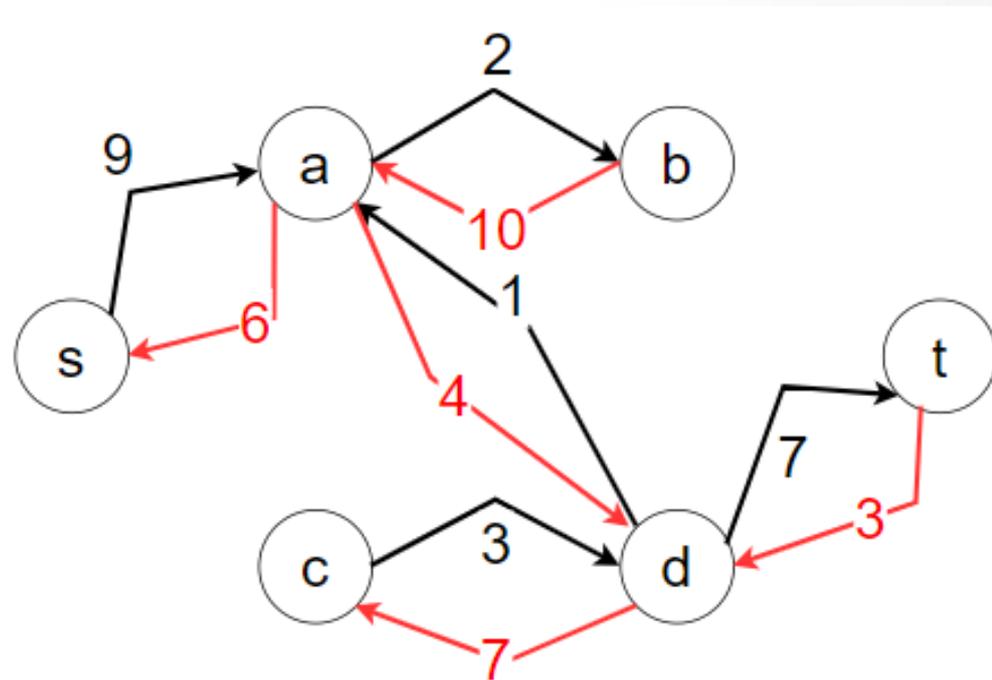
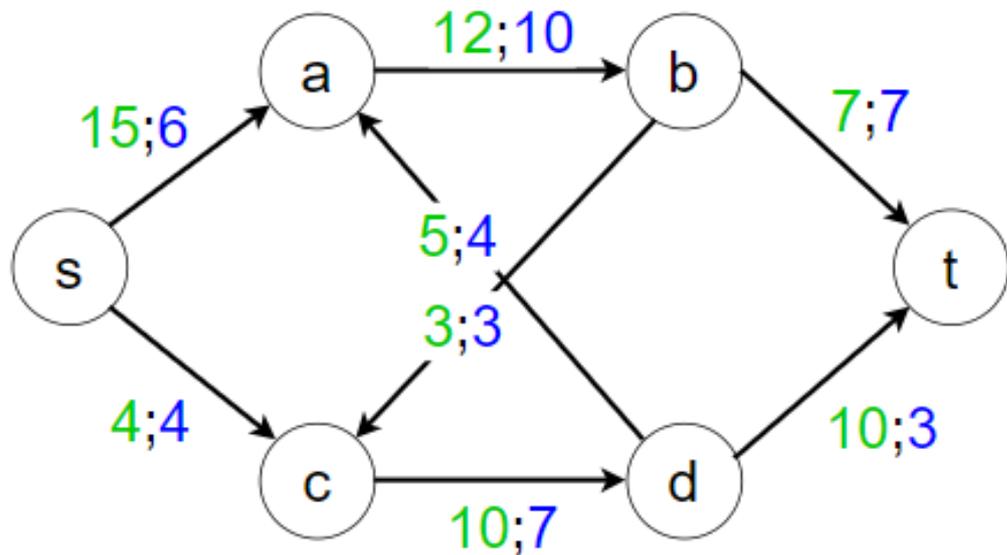
Альтернативная формулировка алгоритма.

Для заданного графа  $G(V, E)$  и потока  $f$  определим *остаточный граф*  $G_f(V, E_f)$ :

- 1) Множество вершин совпадает с вершинами  $G$ .
- 2) Для каждой дуги  $e = (u, v) \in E$  с  $f(e) < c(e)$  в остаточном графе есть *прямая* дуга  $e = (u, v) \in E_f$  с пропускной способностью  $c(e) - f(e)$ .
- 3) Для каждой дуги  $e = (u, v) \in E$  с  $f(e) > 0$  в остаточном графе есть *обратная* дуга  $e = (v, u) \in E_f$  с пропускной способностью  $f(e)$ .

Если для дуги выполняются оба условия (2) и (3), то в  $G_f$  добавляется две дуги – и прямая, и обратная.

# Алгоритм Форда-Фалкерсона

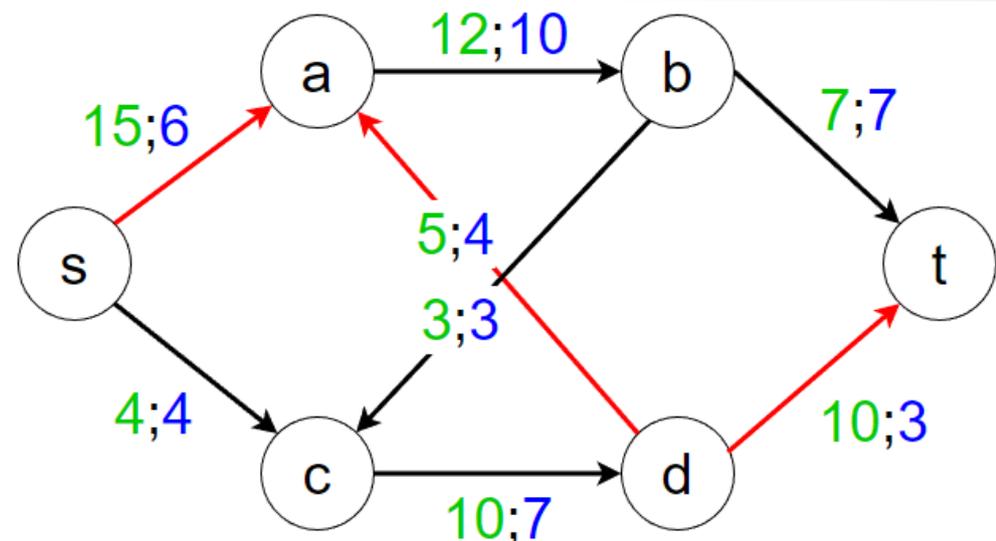
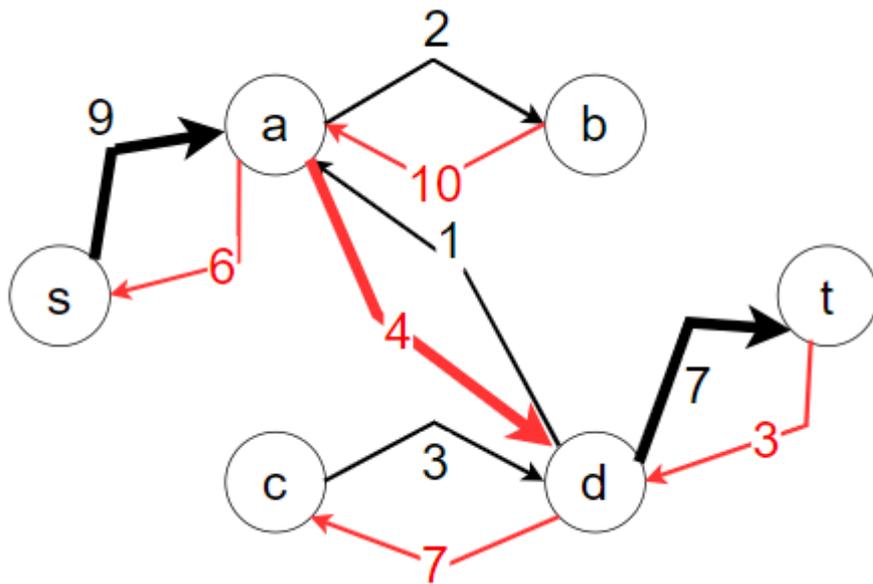


# Алгоритм Форда-Фалкерсона

Одна итерация альтернативной версии алгоритма Форда-Фалкерсона: для текущего потока  $f$ :

- Построить остаточный граф  $G_f$ .
- На  $G_f$  найти ориентированный путь из  $s$  в  $t$ .
- Если такого пути нет, то текущий поток является максимальным.
- Если такой путь  $\pi$  есть, то
  - Для всех дуг пути  $\pi$  вычислить  $\delta = \min c_f(e)$
  - Для всех дуг пути  $\pi$  изменить поток  $f$  на основном графе: для прямых дуг увеличить поток на  $\delta$ , для обратных – уменьшить на  $\delta$ .

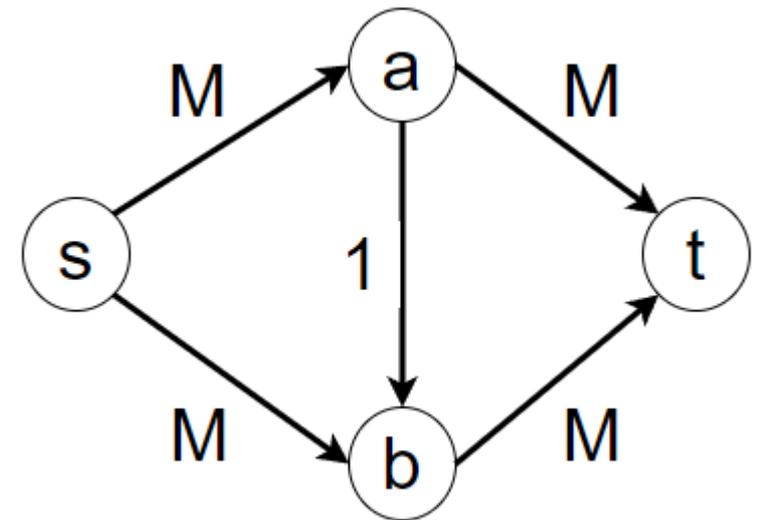
# Алгоритм Форда-Фалкерсона



# Алгоритм Форда-Фалкерсона

Есть ли гарантия, что для любых входных данных  $(G, c)$  алгоритм Форда-Фалкерсона останавливается?

- Если допускаются **иррациональные** пропускные способности, то алгоритм может зацикливаться.
- В вычислительной практике можно считать, что все величины являются рациональными. В этом случае алгоритм гарантированно останавливается (почему?). Но может тратить на это слишком много времени.



# Алгоритм Форда-Фалкерсона

Существуют усовершенствованные версии алгоритма, гарантированно завершающие работу за полиномиальное время:

- Алгоритм Эдмондса-Карпа: при пометке вершин использовать поиск в ширину и выбирать увеличивающую цепь минимальной длины. Сложность:  $O(n^3m)$ .
- Алгоритм Диница:  $O(n^2m)$ .
- Алгоритм Карзанова (усовершенствование алгоритма Диница):  $O(n^3)$ .
- Алгоритм Черкасского:  $O(n^2m^{\frac{1}{2}})$

# Задание 6

# Задание 6

## **Задача «Максимальный поток».**

Для заданного графа с помощью алгоритма Форда-Фалкерсона найти максимальный поток.

### **Формат входных данных**

В первой строке указан список вершин, разделённых пробелами. Первая указанная в списке вершина является источником, последняя – стоком.

Далее следуют строки, представляющие дуги графа (одна строка описывает одну дугу). Каждая строка содержит разделённые пробелами: имя начальной вершин, имя конечной вершины, пропускную способность дуги (число с 2 знаками после запятой).

### **Формат выходных данных**

Первая строка содержит величину максимального потока, представленную в виде числа с 2 знаками после запятой.

Последующие строки описывают поток по дугам в таком же формате, как во входном файле, но вместо пропускной способности указан поток.

# Задание 6

## Пример

input.txt	output.txt
s a b c d t	14.00
s a 15.00	s a 10.00
a b 12.00	a b 10.00
b c 3.00	b c 3.00
c d 10.00	c d 7.00
s c 4.00	s c 4.00
d a 5.00	d a 0.00
b t 7.00	b t 7.00
d t 10.00	d t 7.00