
Сети и потоки

Лектор: Завьялов Олег Геннадьевич
кандидат физико-математических наук, доцент

Сеть:

- подсистема, транспортирующая некий продукт из одной точки в другую. Пример – нефтепровод, электропровод.
- ориентированный граф, ребра которого - трубы между точками системы (вершинами графа).

Каждому ребру $e = (v_i, v_j)$ соответствует положительное целое число $c(e)$, называемое пропускной способностью e .

Если между двумя вершинами не существует ребра, то пропускную способность полагаем равной нулю.

У нефтяных сетей пропускная способность – количество нефти, проходящей через трубу (ребро).

Сеть.

Наличие петель у графа недопустимо.

Если существует ребро из v_i и v_j , то нет ребра из v_j и v_i . (поток продукта только в одну сторону).

Ориентированный граф должен быть связным. Тогда граф называется **простым связным ориентированным графом**.

Особая вершина a , называемая **источником**.

Особая вершина z , называемая **стоком**.

Степень выхода вершины $a = 0$, так что в источнике ничто не втекает.

Степень выхода вершины $z = 0$, так что из стока ничто не вытекает.

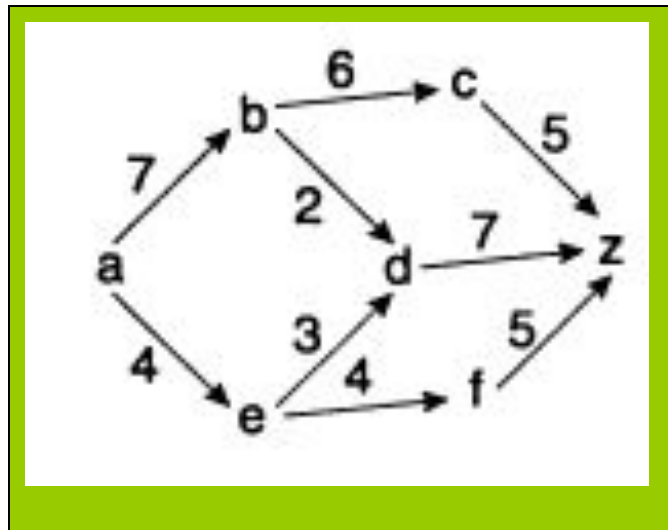
Продукт перевозится из a и имеет место назначения z .

Определение.

Сеть – это ориентированный граф (G, V, E) вместе с весовой функцией $C: E \rightarrow N$ и выделенными вершинами a, z , такими, что

- (1) $\text{indeg}(a) = 0$;
- (2) $\text{outdeg}(z) = 0$.

Например, граф является примером сети, где числа на каждом ребре означают пропускную способность.



Поток.

Для каждого ребра e имеется значение $f(e)$, которое является потоком через конкретное ребро (трубу).

Величина потока не может превысить пропускную способность трубы.

Поток, входящий в вершину, был равен потоку, выходящему из вершины, за исключением вершин a и z - **сохранение потока**.

Пусть $\text{in}(v)$ – множество ребер, для которых v – конечная вершина.

$\text{out}(v)$ – множество ребер, начальная вершина.

$\text{out}(v)$ – множество ребер, выходящих из вершины v ,

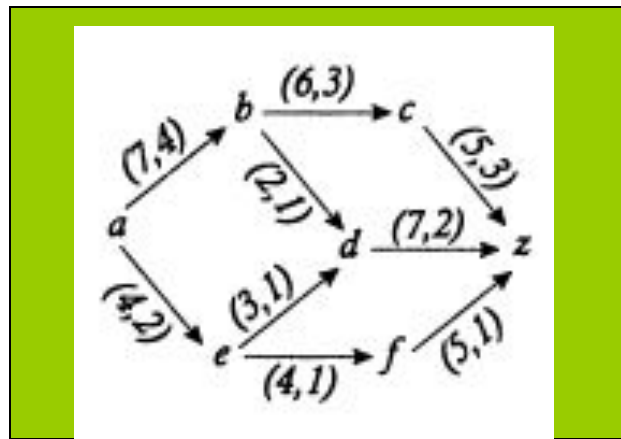
$\text{in}(v)$ – множество ребер, входящих в вершину v .

Определение.

Поток в сети — это функция $f : E \rightarrow \mathbb{N} \cup \{0\}$ такая, что

- (1) для всех $e \in E$, $0 \leq f(e) \leq c(e)$;
- (2) для всех $v \in V$ таких, что $v \neq a, z$, $\sum_{e \in \text{in}(v)} f(e) = \sum_{e \in \text{out}(v)} f(e)$.

Допустим, имеется фиксированный поток. Пусть $\text{поток}(a) = \sum_{e \in \text{out}(a)} f(e)$, т.е. $\text{поток}(a)$ — поток, вытекающий из источника a , и $\text{поток}(z) = \sum_{e \in \text{in}(z)} f(e)$, т.е. $\text{поток}(z)$ — поток, втекающий в сток z . Пример потока в сети, где первый элемент упорядоченной пары на каждом ребре — пропускная способность, а второй

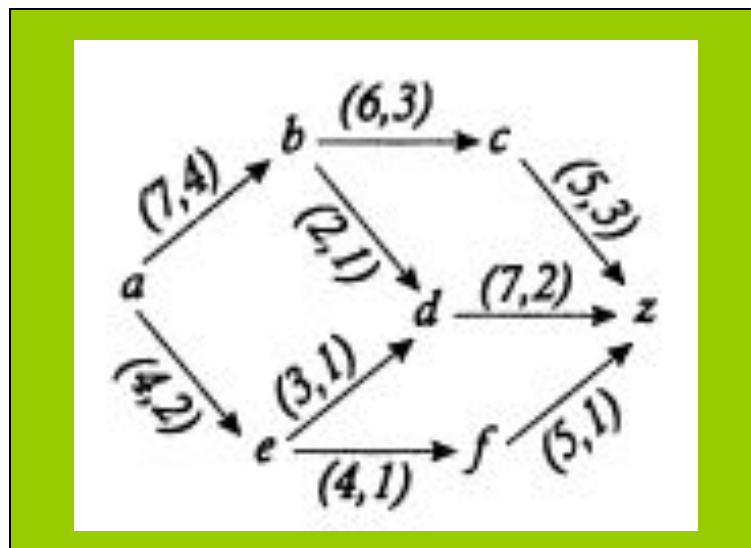


Принцип сохранения потока.

Поток из a равен потоку в z , т.е.

$$\text{поток}(a) = \text{поток}(z)$$

Пример.



вершина	входящий поток	ребро	выходящий поток	ребро
b	4	(a, b)	3	(b, c)
			1	(b, d)
c	3	(b, c)	3	(c, z)
d	1	(e, d)	2	(d, z)
	1	(b, d)		
сумма	9		9	

Пример (продолжение).

Если во множество S добавить вершину a , получим, что разность 15 (выходящий поток) и 9 (входящий поток) равна $6 = \text{поток}(a)$.

вершина	входящий поток	ребро	выходящий поток	ребро
b	4	(a, b)	3	(b, c)
			1	(b, d)
c	3	(b, c)	3	(c, z)
d	1	(e, d)	2	(d, z)
	1	(b, d)		
a	0		4	(a, b)
			2	(a, e)
сумма	9		15	

Пример (продолжение) .

Заметим, что ребра (a, b) , (b, c) и (b, d) присутствуют в колонке для входящего и выходящего потока, потому что обе вершины каждого ребра принадлежат S . При вычитании эти ребра сокращаются, поэтому, удалив их из обеих частей, получаем

вершина	входящий поток	ребро	выходящий поток	ребро
b				
c			3	(c, z)
d	1	(e, d)	2	(d, z)
a	0			
			2	(a, e)
сумма	1		7	

Снова разница входящего и выходящего потока равна 6. Но поскольку ребра, обе вершины которых принадлежат S , удалены, то выходящий поток — это сумма потоков ребер, которые идут из S в T , а входящий поток — это сумма потоков ребер, которые идут из T в S .

Теорема.

Для любого фиксированного потока f

$$\text{поток}(a) = \sum_{e \in \text{out}(a)} f(e) = \sum_{e \in \text{in}(z)} f(e) = \text{поток}(z).$$

Следствие.

Пусть S - подмножество множества V , содержащее a , но не содержащее z , и $T = V - S$. Тогда

$$\sum_{e \in (S,T)} f(e) - \sum_{e \in (T,S)} f(e) = \text{поток}(a) = \text{поток}(z).$$

Определения.

Величина потока f , обозначаемая $\text{val}(f)$, равна

$$\text{поток}(a) = \text{поток}(z) .$$

Пусть S - подмножество множества V и $T = V - S$.

Тогда $\{e : e \in (S, T)\}$ называется **сечением**. Если $a \in S$ и $z \in T$, то сечение называется $a - z$ **сечением**.

Величина $C(S, T) = \sum_{e \in (S, T)}$ называется **пропускной способностью сечения**.

Определения.

Поток f^* в сети называется **максимальным потоком**, если $\text{val}(f^*) \geq \text{val}(f)$ для любого потока f в сети.

$a - z$ сечение (S, T) называется **минимальным сечением**, если $C(S, T)$ не больше пропускной способности любого другого $a - z$ сечения.

Теорема.

Пусть S – подмножество множества V , содержащее a , но не содержащее z , и $T = V - S$. Тогда

$$\text{val}(f) \leq C(S, T).$$

Доказательство:

$$\text{val}(f) = \sum_{e \in (S, T)} f(e) - \sum_{e \in (T, S)} f(e) \leq \sum_{e \in (S, T)} f(e) \leq \sum_{e \in (S, T)} c(e) = C(S, T).$$

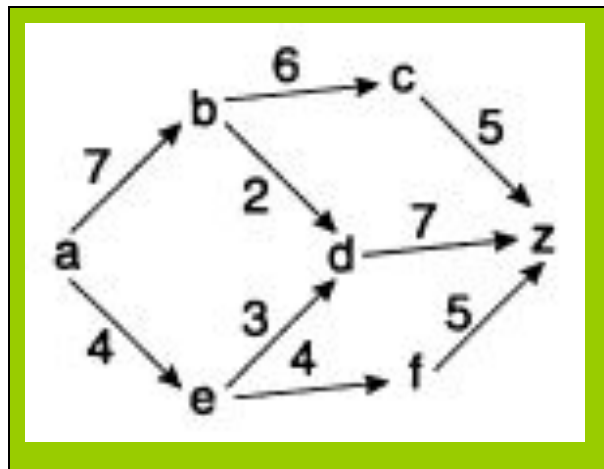
Следствия.

Если $\text{val}(f) = C(S, T)$ для некоторого потока f , а $a - z$ сечения (S, T) , то f – максимальный поток, а C – минимальное сечение.

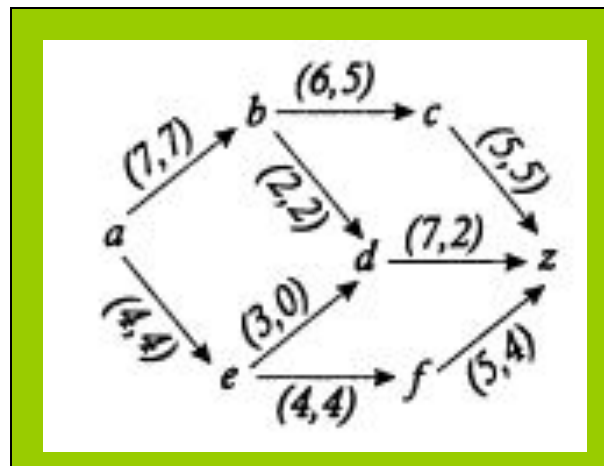
Для некоторого потока f и $a - z$ сечения (S, T) равенство $\text{val}(f) = C(S, T)$ имеет место тогда и только тогда, когда $f(e) = c(e)$ для всех $e \in (S, T)$ и $f(e) = 0$ для всех $e \in (S, T)$.

Способ определения максимального потока сети.

Рассмотрим сеть

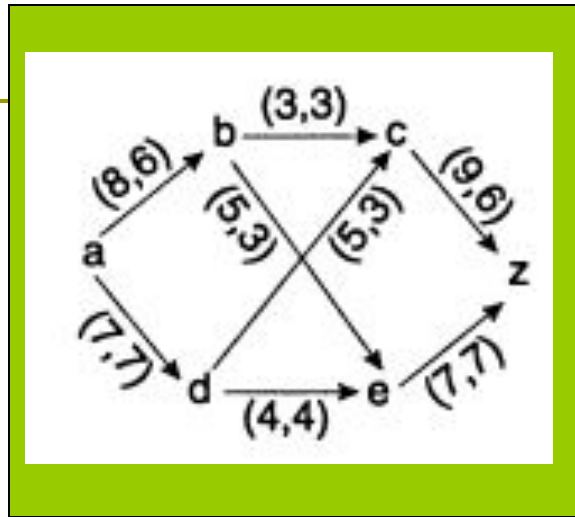


Максимальный поток (не обязательно единственный) легко найти способом

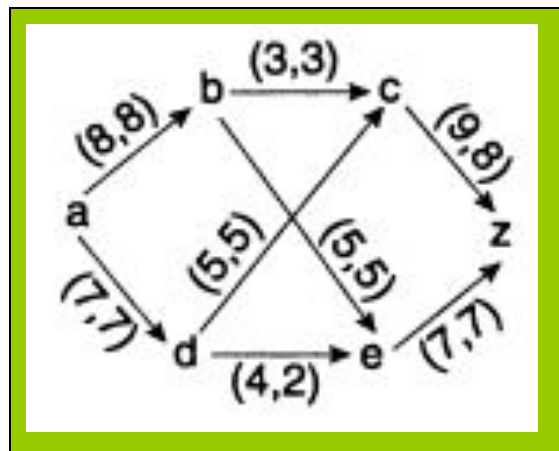


Пример.

Сеть



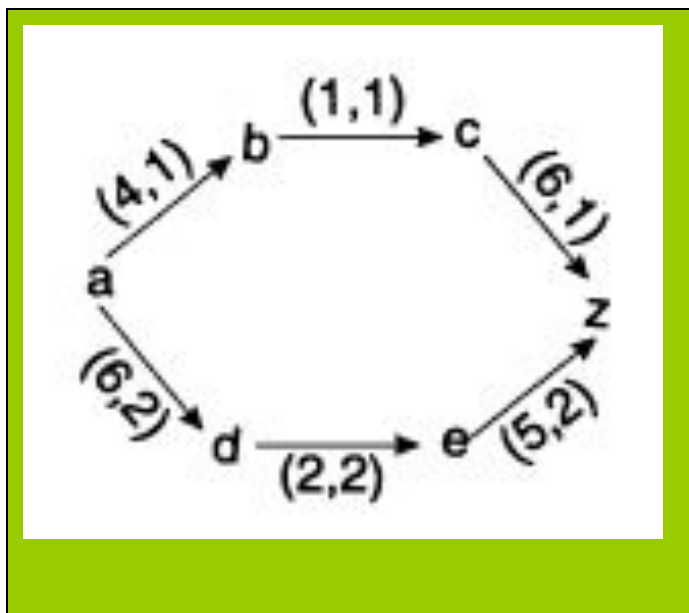
Кажется, что у этой сети максимальный поток, т.к. не существует ориентированного пути, по которому можно увеличить поток. Однако сеть имеет больший поток (максимальный):



Пример.

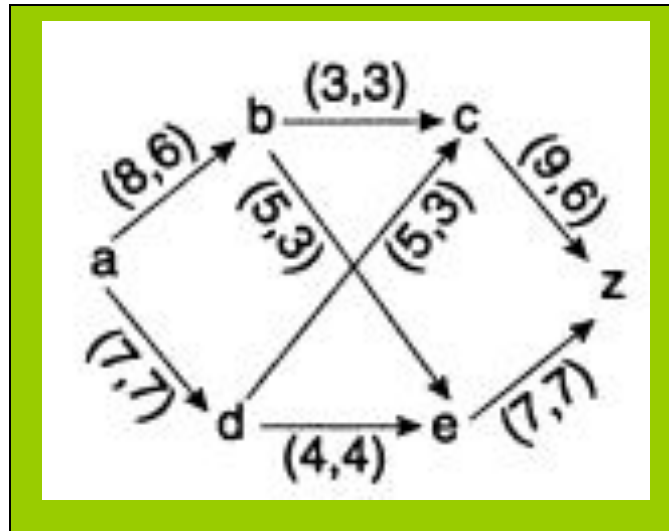
Если поток из источника равен сумме пропускных способностей ребер, выходящих из источника, или поток, втекающий в сток, равен сумме пропускных способностей ребер, входящих в сток, то поток максимальный.

Однако, поток может быть максимальным и без удовлетворения какого-либо из этих признаков. Пример, сеть имеет максимальный поток:



Как находить максимальные в смысле потока сети?

Сформируем пути из a в z , не обращая внимания на направление ребер. Такие пути называют **цепями**. Сеть



Одним из таких путей будет

$$a \xrightarrow{(8,6)} b \xrightarrow{(3,3)} c \xrightarrow{(9,6)} z.$$

Как находить максимальные в смысле потока сети?

Нет возможности увеличить поток по этому пути, т.к. ребро из b в c наполнено до его пропускной способности. То же для цепи

$$a \xrightarrow{(8,6)} b \xrightarrow{(5,3)} e \xrightarrow{(7,7)} z.$$

Цепь

$$a \xrightarrow{(8,6)} b \xrightarrow{(5,3)} e \xleftarrow{(4,4)} d \xrightarrow{(5,3)} c \xrightarrow{(9,6)} z$$

Если увеличить поток на 2 для стрелок, имеющих то же направление, что и цепь, и уменьшить поток на 2 для стрелок, имеющих противоположное направление, получится

$$a \xrightarrow{(8,8)} b \xrightarrow{(5,5)} e \xleftarrow{(4,2)} d \xrightarrow{(5,5)} c \xrightarrow{(9,8)} z,$$

поток увеличивается в 2 раза.

Возникли вопросы.

1) Почему выбираем 2?

Очевидно – поток желательно увеличить как можно больше. Но не превысить пропускную способность ни одного из заданных ребер. Если ребро имеет направление, противоположное цепи, нельзя увеличить поток через это ребро более, чем на текущую величину потока через него, иначе получится отрицательный поток.

Поэтому, если бы не было других ограничений

$$e \xleftarrow{(4,4)} d$$

ограничило бы изменение потока до 4.

Возникли вопросы.

2) Как это влияет на сохранение потока?

Рассмотрим изменение

$$a \xrightarrow{(8,6)} b \xrightarrow{(5,3)} e \quad \text{на} \quad a \xrightarrow{(8,8)} b \xrightarrow{(5,5)} e.$$

Поток, выходящий из вершины b , увеличен на ту же самую величину, что и поток, входящий в вершину b . Поэтому чистый поток через b остается прежним. При изменении

$$b \xrightarrow{(5,3)} e \xleftarrow{(4,4)} d$$

на $b \xrightarrow{(5,5)} e \xleftarrow{(4,2)} d$

поток из вершины b в вершину e увеличен на ту же величину, на которую уменьшен поток из вершины d в вершину e , поэтому чистый поток в вершине e остался тем же.

Продолжение.

Рассмотрим изменение $e \xleftarrow{(4,4)} d \xrightarrow{(5,3)} c$

на $e \xleftarrow{(4,2)} d \xrightarrow{(5,5)} c$

Поток из d в e уменьшен на ту же величину, на которую увеличен поток из d в c . Поэтому чистый поток вершины d остался неизменным.

Процесс увеличения потока в сети.

Формируем цепь из a в z . Если возможно, то увеличиваем поток, определяя наибольшую величину, которую можно добавить к каждому из ребер, имеющих то же самое направление, что и цепь, чтобы поток не превысил пропускную способность, и которую можно вычесть из каждого ребра, имеющего противоположное направление, не создавая отрицательный поток. Поскольку последнее ребро, входящее в z , имеет то же направление, что и цепь, то поток в z возрастает.

С какого потока начать?

Следует начинать с нулевого потока, т.е. когда поток через каждое ребро равен 0, а затем наращивать его. Теперь перейдем к изложению систематического алгоритма для нахождения максимального потока. Каждой вершине поставим в соответствие упорядоченную пару. Первый элемент пары — предшествующая вершина в рассматриваемой цепи предназначена для того, чтобы можно было найти обратный путь. Второй элемент пары — резерв, т.е. величина, на которую можно увеличить поток на каждом ребре вдоль пути, если ориентация ребра совпадает с направлением цепи (такую ориентацию назовем правильной), или уменьшить поток, если ребро ориентировано неправильно. Проще говоря, резерв заданной вершины равен максимальной величине, на которую поток вдоль цепи до этой вершины можно увеличить без нарушения самого потока. Рассмотрим также множество S , содержащее все вершины, не задействованные при построении цепи в z . Если S окажется пустым, до того как будет достигнута вершина z , то больше не будет вершин, которые можно испытать на пути к z , поэтому нет иного пути в z и нет более возможности увеличить поток, значит поток максимален.

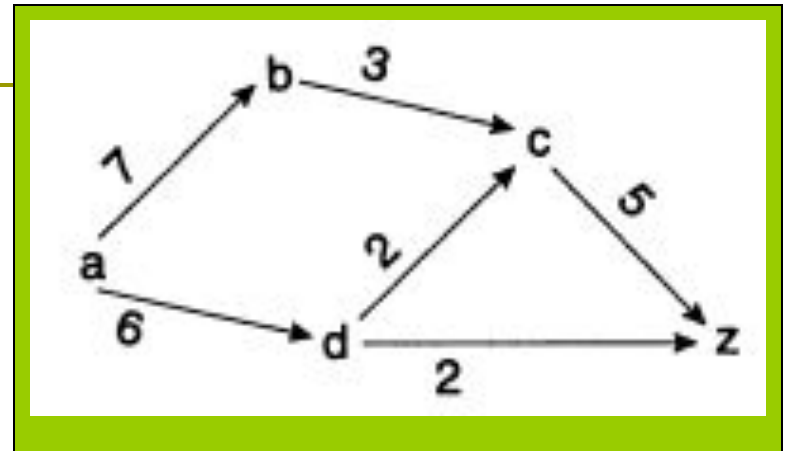
Нахождение максимального потока.

Алгоритм (Форда-Фалкерсона) нахождения максимального потока

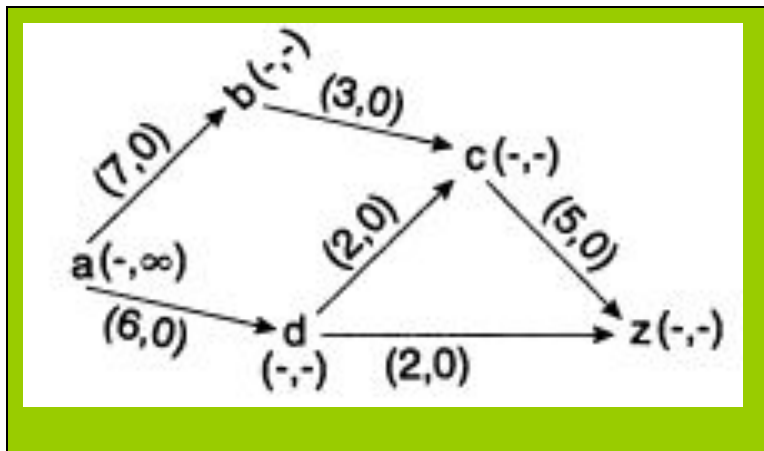
- (1) Установить предшественника каждой вершины и резерв равными символу $-$ (непомечено). Вершина помечена, когда ее резерв не обозначен символом $-$. Установить резерв вершины a равным ∞ , с тем чтобы она не ограничивала резерв других вершин. Положить $S = \{a\}$.
- (2) Если S — пустое множество, то поток максимизирован. Если S не является пустым, то выбираем любой элемент из S и удаляем его. Полагаем этот элемент равным v .
- (3) Если w не помечена, (v, w) является ребром и $f((v, w)) < c((v, w))$. Положить резерв вершины w равным минимуму величины $c((v, w)) - f((v, w))$ и резерва вершины v . Установить предшественника вершины w в v . Если $w \neq z$, добавить w в S .
- (4) Если w не помечена, (w, v) является ребром и $f((v, w)) > 0$. Положить резерв вершины w равным минимуму из $f((v, w))$ и резерва вершины v . Установить предшественника вершины w в v . Добавить w во множество S .
- (5) Если z помечена, то, используя функцию предшествования, вернуться в вершину a и для каждого ребра цепи добавить резерв вершины z к потоку каждого правильно ориентированного ребра, и вычесть резерв z из потока каждого неправильно ориентированного ребра. Вернуться к шагу 1.
- (6) Вернуться к шагу 2.

Пример.

Найти максимальный поток для сети



Шаг 1



Пример (продолжение).

Шаг 2 – проверяем, не пусто ли множество S и выбираем вершину a .

Шаг 3 – полагаем резерв вершины b равным $\min(7, \infty) = 7$.

Устанавливаем вершину a в качестве предшественника вершины b .

Шаг 4 не применяем, возвращаемся в шаг 2.

На шаге 5 видим, что вершина z уже помечена, и, используя функцию предшествования, находим цепь

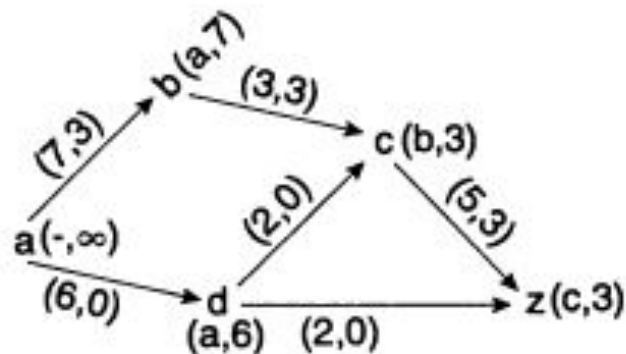
$$a \xrightarrow{(7,0)} b \xrightarrow{(3,0)} c \xrightarrow{(5,0)} z.$$

Добавляя 3 (резерв вершины z) к потоку каждого ребра, получаем

$$a \xrightarrow{(7,3)} b \xrightarrow{(3,3)} c \xrightarrow{(5,3)} z,$$

что дает сеть

Пример (продолжение).



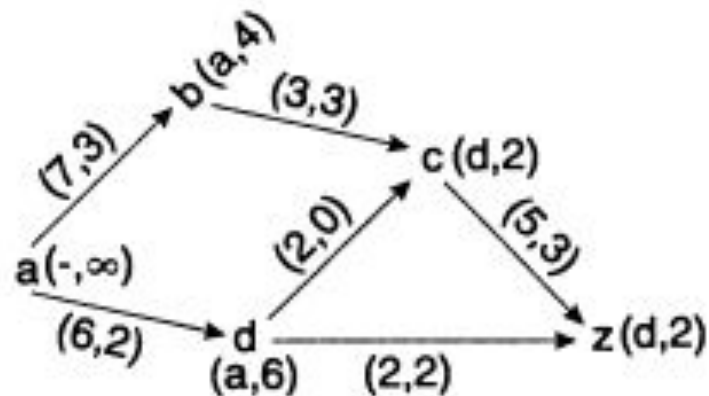
Теперь возвращаемся к шагу 1, где переустанавливаем метки и предшественников и полагаем $S = \{a\}$. На шаге 2 выбираем вершину a из множества S . На шаге 3 устанавливаем резерв вершины b равным $\min(\infty, 7 - 3) = 4$ и устанавливаем a в качестве предшественника вершины b . Помещаем вершину b во множество S . Полагаем также резерв вершины d равным 6 и устанавливаем вершину a в качестве предшественника вершины d . Помещаем вершину d во множество S . Шаги 4 и 5 не применяем, поэтому возвращаемся к шагу 2. Выбираем вершину b из множества S . Поскольку поток из b в c равен пропускной способности из b в c , то вершину c не помечаем. Выбираем вершину d из множества S и продолжаем процесс, после чего получаем цепь

Пример (продолжение).

$$a \xrightarrow{(6,0)} d \xrightarrow{(2,0)} z$$

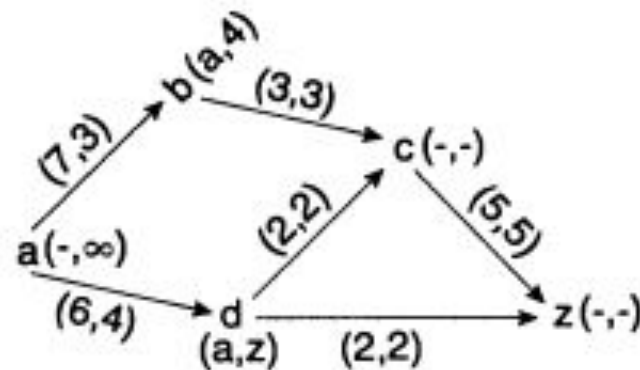
и, добавляя 2 (резерв вершины z) к потоку каждого ребра, получаем цепь

$$a \xrightarrow{(6,2)} d \xrightarrow{(2,2)} z.$$



Пример (продолжение).

Возвращаемся к шагу 1, переустанавливаем метки предшественника и полагаем $S = \{a\}$. На шаге 2 выбираем вершину a из множества S . Как и ранее, на шаге 3 полагаем резерв вершины b равным 4 и устанавливаем вершину a в качестве предшественника вершины b . Помещаем вершину b во множество S . Устанавливаем также резерв вершины d равным $6 - 4 = 2$ и помещаем вершину d во множество S . Возвращаемся к шагу 2. Выбираем вершину b из множества S . Поскольку поток из вершины b в вершину c равен пропускной способности из b в c , пометить c нельзя, поэтому, в конце концов, возвращаемся к шагу 2. Выбираем вершину d из множества S . Поскольку поток из вершины d в вершину c и вершину z равен их пропускным способностям, нельзя пометить c и z . Опять возвращаемся к шагу 2, но множество S пусто. Процесс закончен. Окончательный поток



Теорема.

Алгоритм максимального потока строит максимальный поток для сети.

Теорема.

Поток f на заданной сети N является максимальным тогда и только тогда, когда существует сечение (S, T) такое, что $\text{val}(f) = C(S, T)$.

Последний слайд лекции

!!.