УНИВЕРСИТЕТ ИТМО

УНИВЕРСИТЕТ ИТМО

Дискретная математика

Преподаватель:

к.ф.-м.н., доцент НОЦ математики

Фалеева Мария Петровна

mpfaleeva@itmo.ru

Санкт-Петербург, 2022

Лекция 7. Графы

Связность. Компоненты связности

- Пусть $G = G(V,E)$ – граф.
- Граф $G'(V',E')$ называется подграфом (или частью) графа $G(V,E)$ (обозначается $G' \subset G$), если $V' \subset V$ и $E' \subset E$.
- Если $V' = V$, то G' называется остовным подграфом G
- Говорят, что две вершины в графе связаны, если существует соединяющая их (простая) цепь.
- Граф, в котором все вершины связаны, называется связным.
- Подграф G' графа G называется компонентой графа G (компонентой связности), если G' - максимальный связный подграф графа G.

Связность. Компоненты связности

- Отношение связанности вершин является эквивалентностью. Классы эквивалентности по отношению связанности являются компонентами связности графа.
- Число компонент связности графа G обозначается $k(G)$.
- Граф G является связным тогда и только тогда, когда $k(G) = 1$.
- Если $k(G) > 1$, то G – несвязный граф.
- Граф, состоящий только из изолированных вершин называется вполне несвязным.

Связность. Компоненты связности

- Пример

Графы на рисунках 1 и 2 являются компонентами графа, представленного на рисунке 3

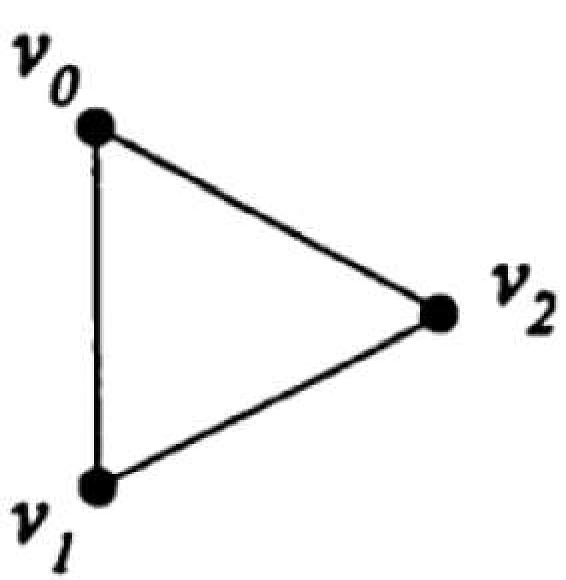


Рисунок 1

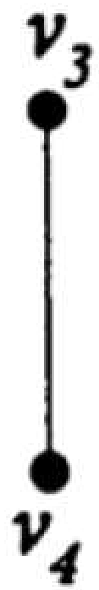


Рисунок 2

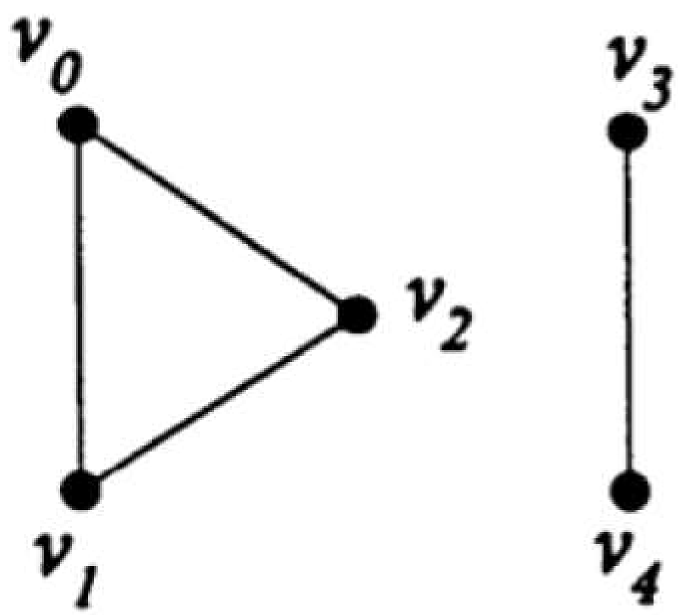


Рисунок 3

Связность. Компоненты связности

- Пусть $G(V, E)$ – орграф, v_1 и v_2 - его узлы. Говорят, что два узла, v_1 и v_2, сильно связаны в орграфе G, если существует путь (ориентированная цепь) как из v_1 в v_2, так и из v_2 в v_1.
- Компонента сильной связности (КСС) орграфа G – это его максимальный сильно связный подграф.
- Вершина графа называется точкой сочленения, или разделяющей вершиной, если её удаление увеличивает число компонент связности.
- Мостом называется ребро, удаление которого увеличивает число компонент связности.
- Блоком называется связный граф, не имеющий точек сочленения.

Связность. Компоненты связности

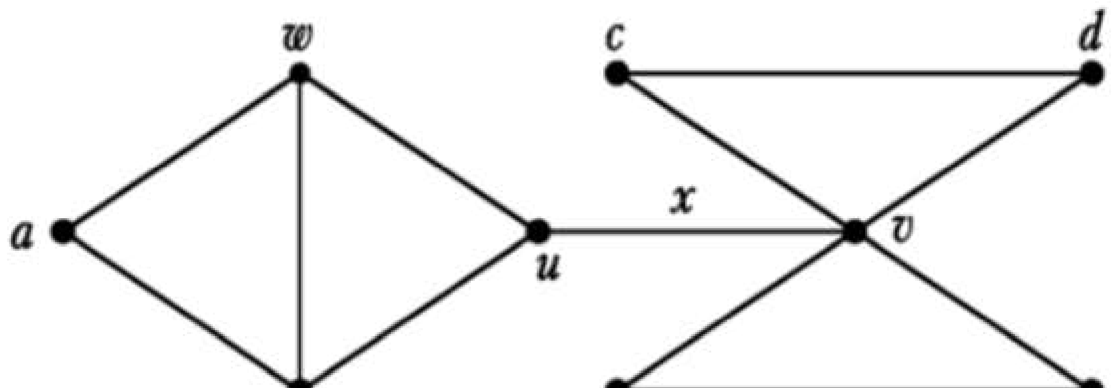
- Пример

В графе, диаграмма которого представлена ниже:

1) вершины u, v – точки сочленения, и других точек сочленения нет;

2) ребро x – мост, и других мостов нет;

3) подграфы $\{a, b, w\}, \{b, u, w\}, \{a, b, u, w\}, \{c, d, v\}, \{e, f, v\}$ – блоки и других блоков нет.



Связность. Компоненты связности

- Теорема

Любой граф представляется в виде объединения непересекающихся связных (сильно связных) компонент.

Разложение графа на связные (сильно связные) компоненты определяются однозначно.

Матрицы связности и достижимости

- Вершина v_j достижима из вершины v_i, если существует маршрут, соединяющий вершины $v_i, v_j \in V$.
- Достижимость из всех исходных вершин графа описывается матрицей достижимости **R** (reach), элемент которой $r_{ij}, i, j = \overline{1, n}$:

$$r_{ij} = \begin{cases} 1, \text{если вершина } v_j \text{ достижима из } v_i; \\ 0, \text{иначе.} \end{cases}$$

- Поскольку каждая вершина достижима из себя самой, в матрице достижимости $r_{ii} = 1$.
- Для неориентированного графа матрица достижимости называется также матрицей связности.

Матрицы связности и достижимости

Вычисление матрицы достижимости

Пусть $P(G)$ - матрица смежности вершин графа $G(V,E)$, а $B = E + P + P^2 + \dots + P^n$.

Тогда элементы матрицы достижимости $C = (c_{ij}), i, j = \overline{1,n}$ вычисляются следующим образом:

$$c_{ij} = \begin{cases} 1, \text{если } b_{ij} \neq 0, \\ 0, \text{если} b_{ij} = 0. \end{cases}$$

Матрица контрдостижимости

- Матрица $L = \left(l_{ij}\right), i, j = \overline{1, n}$ такая, что $L = C^T$, т.е.

$$l_{ij} = \begin{cases} 1, \text{если вершина } x_i \text{ достижима из вершины } x_j, \\ 0, \text{если вершина } x_i \text{ не достижима из вершины } x_j, \end{cases}$$

называется <u>матрицей контрдостижимости</u>.

Нахождение сильных компонент графа

- Вычислим матрицу компонент сильной связности:

$$F = C * L,$$

где операция $*$ означает поэлементное произведение матриц C и L:

$$f_{ij} = c_{ij} \cdot l_{ij}.$$

- Элемент f_{ij} матрицы F равен единице тогда и только тогда, когда вершины x_i и x_j взаимно достижимы, т.е. сильно связанны.
- Сильная компонента орграфа, содержащая вершину x_i, состоит из элементов x_j, для которых $f_{ij} = 1$.

Пример

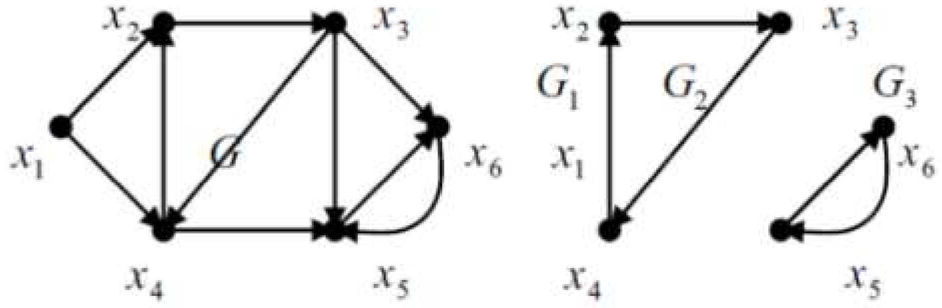


Рис. 3.12.

$$P=\begin{pmatrix}0&1&0&1&0&0\\0&0&1&0&0&0\\0&0&0&1&1&1\\0&1&0&0&1&0\\0&0&0&0&0&1\\0&0&0&0&1&0\end{pmatrix},\quad P^2=\begin{pmatrix}0&1&0&0&1&0\\0&0&0&1&1&1\\0&1&0&0&2&1\\0&0&1&0&0&1\\0&0&0&0&1&0\\0&0&0&0&0&1\end{pmatrix},\quad P^3=\begin{pmatrix}0&0&1&0&0&1\\0&1&0&0&2&1\\0&0&1&0&1&2\\0&0&0&1&2&1\\0&0&0&0&0&1\\0&0&0&0&1&0\end{pmatrix},$$

$$P^4=\begin{pmatrix}0&0&0&1&2&1\\0&0&1&0&1&2\\0&0&0&1&3&2\\0&1&0&0&2&2\\0&0&0&0&1&0\\0&0&0&0&0&1\end{pmatrix},\quad P^5=\begin{pmatrix}0&1&0&0&2&2\\0&0&0&1&3&2\\0&1&0&0&3&3\\0&0&1&0&2&2\\0&0&0&0&0&1\\0&0&0&0&1&0\end{pmatrix},\quad P^6=\begin{pmatrix}0&0&1&0&2&2\\0&1&0&0&3&3\\0&0&1&0&3&3\\0&0&0&1&3&3\\0&0&0&0&1&0\\0&0&0&0&0&1\end{pmatrix},$$

Пример

$$B = E + P + P^2 + P^3 + P^4 + P^5 + P^6 = \begin{pmatrix} 1 & 3 & 2 & 2 & 7 & 6 \\ 0 & 3 & 2 & 2 & 10 & 9 \\ 0 & 2 & 3 & 2 & 13 & 12 \\ 0 & 2 & 2 & 3 & 10 & 9 \\ 0 & 0 & 0 & 0 & 4 & 3 \\ 0 & 0 & 0 & 0 & 3 & 4 \end{pmatrix}, \qquad C = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

$$L = C^{\mathrm{T}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad F = C * L = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}.$$

По матрице F легко определить состав вершин трех подграфов, образующих сильно связные компоненты исходного графа.

Расстояние между вершинами, ярусы и диаметр графа

- Расстоянием между вершинами u и v (обозначается $d(u,v)$) называется длина кратчайшей цепи $\langle u,v\rangle$:
$$d(u,v) \overset{\text{def}}{=} \min_{\{\langle u,v\rangle\}} |\langle u,v\rangle|$$
- Замечание. Если $\neg\exists\ (\langle u,v\rangle)$, то по определению $d(u,v) \overset{\text{def}}{=} +\infty$.
- Множество вершин, находящихся на заданном расстоянии n от вершины v (обозначение $D(v,n)$), называется ярусом:
$$D(v,n) \overset{\text{def}}{=} \{u \in V \mid d(v,u) = n\}.$$
- Ясно, что множество вершин V всякого связного графа однозначно разбивается на ярусы относительно данной вершины.

Эксцентриситет и центр

- Эксцентриситетом $e(v)$ вершины v в связном графе $G(V,E)$ называется максимальное из расстояний от вершины v до других вершин графа G:
$$e(v) \stackrel{\text{def}}{=} \max_{u \in V} d(v,u)$$
- Радиусом $R(G)$ графа G называется наименьший из эксцентриситетов вершин:
$$R(G) \stackrel{\text{def}}{=} \min_{v \in V} e(v)$$
- Диаметром графа G называется наибольший из эксцентриситетов графа. Длина диаметра обозначается $D(G)$:
$$D(G) \stackrel{\text{def}}{=} \max_{v \in V} e(v)$$

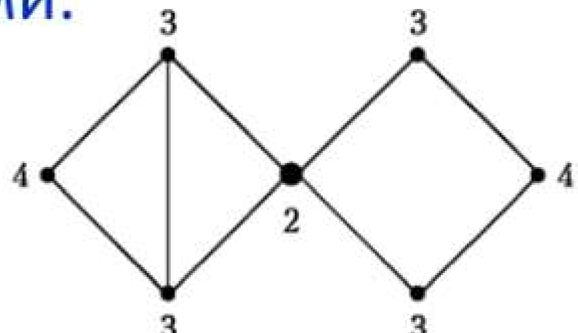
Эксцентриситет и центр

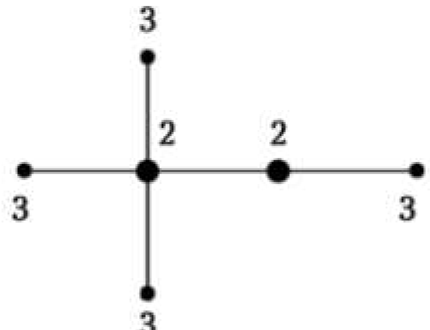
- Вершина v называется центральной, если ее эксцентриситет совпадает с радиусом графа, $e(v) = R(G)$.
- Множество центральных вершин называется центром графа и обозначается $C(G)$:

$$C(G) \stackrel{\text{def}}{=} \{v \in V \mid e(v) = R(G)\}$$

- Пример

На рисунках указаны эксцентриситеты вершин и центры двух графов. Вершины, составляющие центр, выделены жирными точками.





Изоморфизм графов

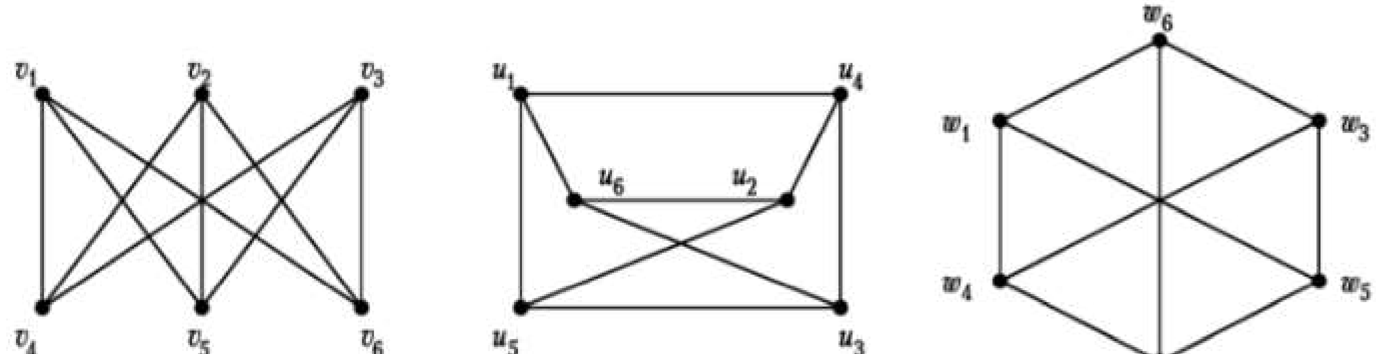
- Два графа $G_1(V_1, E_1)$ и $G_2(V_2, E_2)$ изоморфны (обозначается $G_1 \sim G_2$, или $G_1 = G_2$), если существует биекция $h\colon V_1 \to V_2$, сохраняющая смежность:

$$e_1 = (u, v) \in E_1 \Leftrightarrow e_2 = (h(u), h(v)) \in E_2.$$

- Теорема

  Изоморфизм графов есть отношение эквивалентности.

- Пример. Три внешне различные диаграмм, приведенные ниже, являются диаграммами одного и того же графа $K_{3,3}$.



Изоморфизм графов

- Теорема

Графы изоморфны тогда и только тогда, когда их матрицы смежности вершин получаются друг из друга одновременными перестановками строк и столбцов (т.е. одновременно с перестановкой i-й j-й строк переставляются i-й j-й столбцы).

- Теорема

Графы (орграфы) изоморфны тогда и только тогда, когда их матрицы инциденций получаются друг из друга произвольными перестановками строк и столбцов.

Поиск кратчайших путей. Алгоритм Дейкстры

Алгоритм нахождения кратчайшего пути между двумя заданными вершинами в орграфе.

Ограничение – веса дуг должны быть положительными.

- Пусть $G = \{S, U, \Omega\}$ - ориентированный граф со взвешенными дугами.
- Пусть вершина s – начало пути, t – конец пути.
- $d(x_i)$ – оценка длины (вес) кратчайшего пути от s к x_i.
- Постоянные метки помечаются сверху звёздочкой.
- $\tilde{x}$ - текущая вершина.

Поиск кратчайших путей. Алгоритм Дейкстры

- Этап 1. Нахождение длины кратчайшего пути.

1) Полагаем $d(s) = 0^*$. Для остальных вершин $x_i \in S, x_i \neq s$, полагаем $d(x_i) = \infty$ и считаем эти метки временными. $\tilde{x} = s$.

2) Для каждой вершины x_i с временной меткой, непосредственно за следующей за вершиной $\tilde{x}$, меняем ее метку:

$$d_{\text{нов.}}(x_i) = \min\{d_{\text{стар.}}(x_i), d(\tilde{x}) + \omega(\tilde{x}, x_i)\}.$$

3) Из всех вершин с временными метками выбираем вершину x_j^* с наименьшим значением метки:

$$d(x_j^*) = \min\{d(x_j) \mid x_j \in S, d(x_j) - \text{временная}\}$$

Превращаем эту метку в постоянную и полагаем $\tilde{x} = x_j^*$.

4) Если $\tilde{x} = t$, то $d(\tilde{x})$ – длина кратчайшего пути от s к t. В

Поиск кратчайших путей. Алгоритм Дейкстры

- Этап 2. Построение самого кратчайшего пути.

5) Среди вершин, непосредственно предшествующих вершине $\tilde{x}$ с постоянными метками, находим вершину x_i, удовлетворяющую соотношению

$$d(\tilde{x}) = d(x_i) + \omega(x_i, \tilde{x}).$$

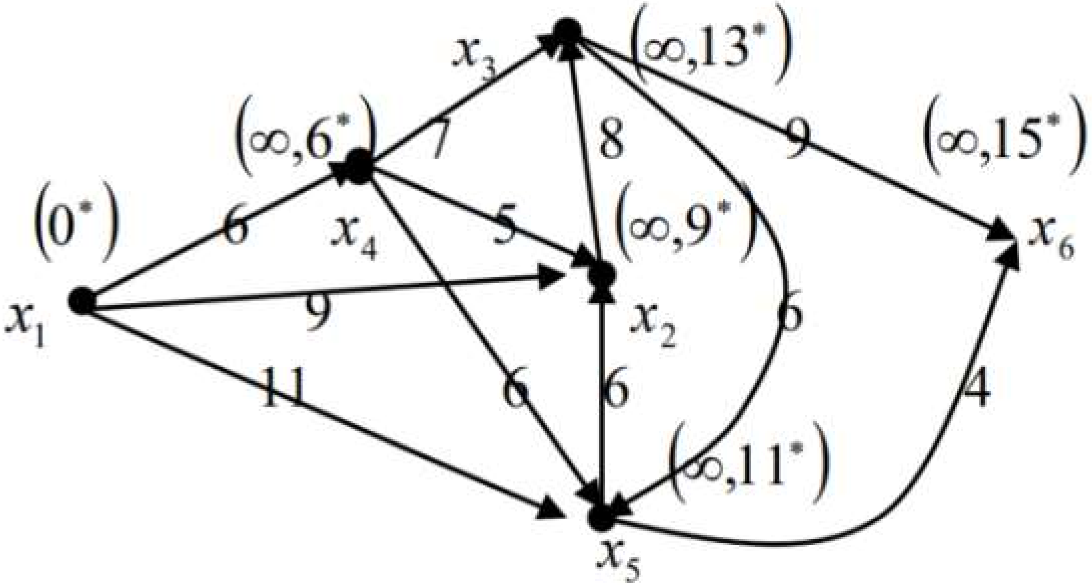
Включаем дугу $(x_i, \tilde{x})$ в искомый путь и полагаем $\tilde{x}= x_i$.

6) Если $\tilde{x} = s$, то кратчайший путь найден – его образует последовательность дуг, полученных на пятом шаге и выстроенных в обратном порядке. В противном случае возвращаемся к пятому шагу.

Поиск кратчайших путей. Алгоритм Дейкстры

- Пример.

$s = x_1; t = x_6.$



$$\Omega = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{bmatrix} 0 & 9 & 0 & 6 & 11 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 9 \\ 0 & 5 & 7 & 0 & 6 & 0 \\ 0 & 6 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Поиск кратчайших путей. Алгоритм Дейкстры

- * - постоянные метки (пройденные вершины)

_ - смежные, непройденные вершины;

$\tilde{x}$	S	$d[x_1$	x_2	x_3	x_4	x_5	$x_6]$
x_1	$\{\}$	$[0^*$	∞	∞	∞	∞	$\infty]$
x_4	$\{x_2, x_4, x_5\}$	$[0^*$	9	∞	6^*	11	$\infty]$
x_2	$\{x_2, x_3, x_5\}$	$[0^*$	9^*	13	6^*	11	$\infty]$
x_5	$\{x_3\}$	$[0^*$	9^*	13	6^*	11^*	$\infty]$
x_3	$\{x_6\}$	$[0^*$	9^*	13^*	6^*	11^*	$15]$
x_6	$\{x_6\}$	$[0^*$	9^*	13^*	6^*	11^*	$15^*]$

$$
\begin{array}{c c}
 & \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} &
\begin{bmatrix}
0 & 9 & 0 & 6 & 11 & 0 \\
0 & 0 & 8 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 6 & 9 \\
0 & 5 & 7 & 0 & 6 & 0 \\
0 & 6 & 0 & 0 & 0 & 4 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

- Кратчайший путь: $x_1 \to x_5 \to x_6$ и его длина (вес) равна 15.

Поиск кратчайших путей. Алгоритм Беллмана-Мура.

Алгоритм нахождения кратчайшего пути между двумя заданными вершинами в орграфе.

Ограничение – в графе не должно быть ориентированных циклов отрицательного веса

Отличие от алгоритма Дейкстры: нет постоянных меток, формирование очереди вершин

Поиск кратчайших путей. Алгоритм Беллмана-Мура.

Этап 1. Нахождение длины кратчайших путей от вершины s до всех остальных вершин графа.

1) Присвоение начальных значений. $d(s) = 0, d\left(x_j\right) = \infty$, $x_j \in S, \tilde{x} = s, Q = \{\tilde{x}\}$ – множество вершин в очереди.

2) Удаляем из очереди Q вершину, находящуюся в самом начале очереди.

Для каждой вершины x_i, непосредственно достижимой из $\tilde{x}$, корректируем ее метку по формуле

$$d_{\text{нов.}}(x_i) = \min\left\{d_{\text{стар.}}(x_i), d(\tilde{x}) + \omega(\tilde{x}, x_i)\right\}.$$

Если $d_{\text{нов.}}(x_i) < d_{\text{стар.}}(x_i)$, то корректируем очередь вершин, иначе продолжаем перебор вершин и корректировку временных меток.

Поиск кратчайших путей

Корректировка очереди.

Если x_i не была ранее в очереди и не находится в ней в данный момент, то вершину x_i ставим в конец очереди. Если же x_i уже была когда-нибудь в очереди или находится там в данный момент, то переставляем ее в начало очереди.

3) Если $Q \neq \emptyset$, то возвращаемся к началу второго шага, если $Q = \emptyset$, то первый этап закончен.

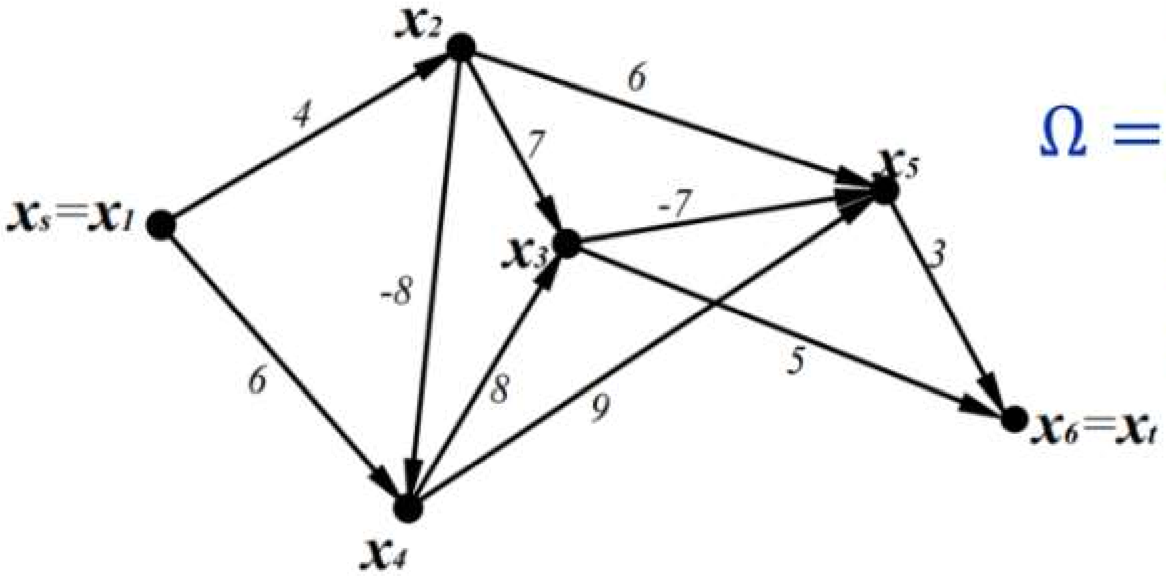
- Этап 2. Построение кратчайшего пути.

Совпадает со 2 этапом в алгоритме Дейкстры.

Поиск кратчайших путей

- Пример. Беллмана-Мура.

$s = x_1; t = x_6.$



$$\Omega = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{bmatrix} 0 & 4 & 0 & 6 & 0 & 0 \\ 0 & 0 & 7 & -8 & 6 & 0 \\ 0 & 0 & 0 & 0 & -7 & 5 \\ 0 & 0 & 8 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Поиск кратчайших путей

$\tilde{x}$	$\tilde{S}$	d $[x_1,$	$x_2,$	$x_3,$	$x_4,$	$x_5,$	$x_6]$	Q
x_1	$\{x_2, x_4\}$	[0	4	∞	6	∞	∞]	$[x_2, x_4]$
x_2	$\{x_3, x_4, x_5\}$	[0	4	11	-4	10	∞]	$[x_4, x_3, x_5]$
x_4	$\{x_3, x_5\}$	[0	4	8	-4	5	∞]	$[x_5, x_3]$
x_5	$\{x_6\}$	[0	4	8	-4	5	8]	$[x_3, x_6]$
x_3	$\{x_5, x_6\}$	[0	4	8	-4	-3	8]	$[x_5, x_6]$
x_5	$\{x_6\}$	[0	4	8	-6	-3	0]	$[x_6]$

$$\begin{bmatrix} 0 & 4 & 0 & 6 & 0 & 0 \\ 0 & 0 & 7 & -8 & 6 & 0 \\ 0 & 0 & 0 & 0 & -7 & 5 \\ 0 & 0 & 8 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Кратчайший путь: $x_1 \to x_2 \to x_4 \to x_3 \to x_5 \to x_6$.

Нахождение максимального потока в сети

- Сеть – математическая модель физических систем, в которых некоторые объекты текут, движутся или транспортируются по системе каналов (дуг сети) ограниченной пропускной способности.
- Примеры: поток автомобильного транспорта по сети автодорог, поток воды в городской сети водоснабжения, поток электрического тока в электросети, поток телефонных по каналам связи, поток программ в вычислительной сети.
- Ограниченная пропускная способность означает, что интенсивность перемещения соответствующих предметов по каналу ограничена сверху определенной величиной.

Нахождение максимального потока в сети

- Граф $G = (S, U)$ является сетью, если

1) G – связный граф без петель;

2) существует ровно одна вершина, не имеющая предшествующих (источник - s);

3) существует ровно одна вершина, не имеющая последующих (сток - t);

4) каждой дуге $(x_i, x_j) \in U$ поставлено в соответствие неотрицательное число $c(x_i, x_j)$, $c(x_i, x_j) \in \Omega$, называемое пропускной способностью дуги.

Нахождение максимального потока в сети

- Функция $\varphi(x_i, x_j)$, определенная на множестве дуг сети $G = (S, U, \Omega)$, называется потоком, если

1. $\forall (x_i, x_j) \in U \qquad 0 \leq \varphi(x_i, x_j) \leq c(x_i, x_j)$
2. $\sum_{x_i \in S_{\text{пр}}(x_j)} \varphi(x_i, x_j) = \sum_{x_k \in S_{\text{сл}}(x_j)} \varphi(x_j, x_k) \; \forall \; x_j \in S$ и $x_j \notin \{s, t\}$.

   (условие сохранения потока: в промежуточных вершинах потоки не создаются и не исчезают)

- Величина $\triangle(x_i, x_j) = c(x_i, x_j) - \varphi(x_i, x_j)$ называется остаточной пропускной способностью дуги (x_i, x_j).
- Если $\varphi(x_i, x_j) = c(x_i, x_j)$, то дуга называется насыщенной.

Нахождение максимального потока в сети

- Разрез может быть определен как множество дуг, исключение которых из сети отделило бы некоторое множество узлов от остальной сети.
- Предположим, что множество вершин сети S разбито на два непустых непересекающихся подмножества $S = S' \cup S''$ и $S' \cap S'' = \emptyset$.
- Множество дуг, начала которых лежат в S', а концы в S'', называется ориентированным разрезом и обозначается $(S' \to S'')$. Следовательно,

$$(S' \to S'') = \{(x_i, x_j) | x_i \in S', x_j \in S''\}.$$

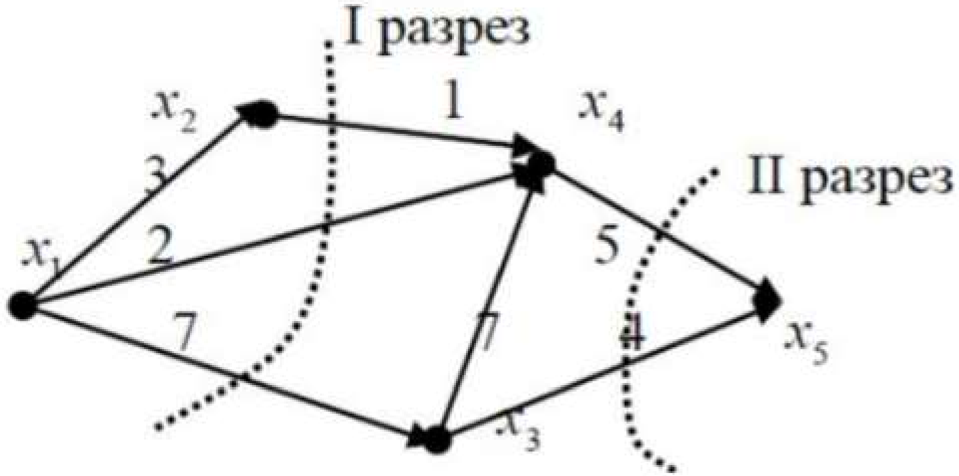
Нахождение максимального потока в сети

- Пропускной способностью или величиной разреза $(S' \to S'')$ называется сумма пропускных способностей входящих в него дуг, то есть

$$c(S' \to S'') = \sum_{x_i \in S', x_j \in S''} c(x_i, x_j)$$

Нахождение максимального потока в сети

- Пример



- При разрезе I вершины оказались разбиты на подмножества $S' = \{x_1, x_2\}$ и $S'' = \{x_3, x_4, x_5\}$, а ребрами, образующими разрез стали ребра $(x_1, x_3), (x_1, x_4), (x_2, x_4)$.
- При разрезе II $S' = \{x_1, x_2, x_3, x_4\}$ и $S'' = \{x_5\}$, а разрез образуют ребра $(x_3, x_5), (x_4, x_5)$.

Теорема Форда-Фалкерсона

- Теорема. Для любой сети с одним источником и одним стоком величина максимального потока в сети от источника к стоку равна пропускной способности минимального разреза.
- Алгоритм нахождения максимального потока

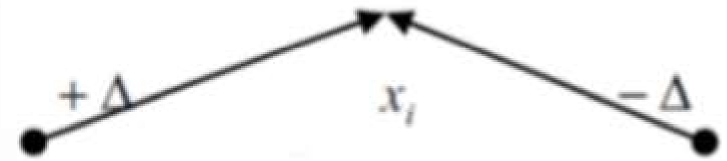
Рассмотрим произвольный маршрут (неориентированный путь) из s в t. Дуги, образующие этот маршрут, естественным образом делятся на два типа: прямые (ориентированные от s к t) и обратные (ориентированные от t к s) .

Пусть существует путь, в котором прямые дуги не насыщены, а потоки на обратных дугах положительны.

Пусть $\triangle_1$ - минимальная из остаточных пропускных способностей прямых дуг, а $\triangle_2$ - минимальная из величин

Теорема Форда-Фалкерсона

Тогда поток в сети можно увеличить на величину $\triangle = \min\{\triangle_1, \triangle_2\}$, прибавляя $\triangle$ к потокам на прямых дугах и вычитая $\triangle$ из потоков на обратных дугах.



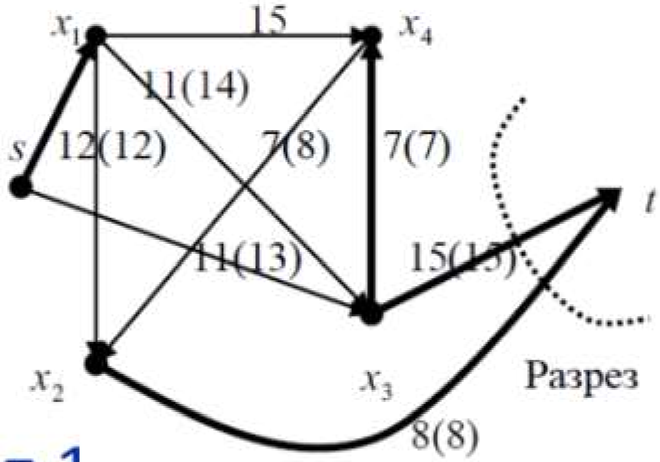
Очевидно, что при этом условие баланса (условие сохранение потока)

$$\sum_{x_i \in S_{\text{пр}}(x_j)} \varphi(x_i, x_j) = \sum_{x_k \in S_{\text{сл}}(x_j)} \varphi(x_j, x_k)$$

для узлов, входящих в рассматриваемый маршрут не нарушится.

Теорема Форда-Фалкерсона

- Пример. Пропускные способности дуг заданы следующей матрицей. Построить максимальный поток от s к t и указать минимальный разрез, отделяющий s от t.



$$\Omega = \begin{array}{c} \\ s \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ t \end{array} \begin{array}{c} \begin{matrix} s & x_1 & x_2 & x_3 & x_4 & t \end{matrix} \\ \begin{bmatrix} 0 & 12 & 0 & 13 & 0 & 0 \\ 0 & 0 & 11 & 14 & 15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 7 & 15 \\ 0 & 0 & 8 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Этап 1.

Путь $s \xrightarrow{12} x_1 \xrightarrow{14} x_3 \xrightarrow{15} t,\ \delta = min(12,14,15) = 12$. Увеличим по этому пути поток до 12 единиц, ребро (s, x_1) становится насыщенным. Получили $s \xrightarrow{12(12)} x_1 \xrightarrow{12(14)} x_3 \xrightarrow{12(15)} t,$

Теорема Форда-Фалкерсона

Поставим величину потока на дугах (x_1, x_3) и (x_3, t). Путь $s \xrightarrow{13} x_3 \xrightarrow{12(15)} t$. $\delta = min(13, 15 - 12) = 3$. Поток можно увеличить на три единицы. Получили $s \xrightarrow{3(13)} x_3 \xrightarrow{15(15)} t$. Дуга (x_3, t) станет насыщенной.

Рассмотрим путь $s \xrightarrow{3(13)} x_3 \xrightarrow{7} x_4 \xrightarrow{8} x_2 \xrightarrow{8} t$. Можно увеличить поток на семь единиц, дуга (x_3, x_4) станет насыщенной, потоки на дугах примут вид: $s \xrightarrow{3(13)} x_3 \xrightarrow{7(7)} x_4 \xrightarrow{7(8)} x_2 \xrightarrow{7(8)} t$. Больше путей нет.

Теорема Форда-Фалкерсона

Этап 2.

Рассмотрим теперь маршруты, содержащие противоположные дуги.

Маршрут $s \xrightarrow{10(13)} x_3 \xleftarrow{12(14)} x_1 \xrightarrow{11} x_2 \xrightarrow{7(8)} t$. Поток можно увеличить на единицу на дуге (x_2, t).

Тогда потоки по дугам этого маршрута станут такими $s \xrightarrow{11(13)} x_3 \xleftarrow{11(14)} x_1 \xrightarrow{1(11)} x_2 \xrightarrow{8(8)} t$. Дуга (x_2, t) стала насыщенной.

Больше маршрутов нет. Поток максимален. Делаем разрез вокруг t по насыщенным дугам и получаем его величину 15+8=23 единицы.

Определения

- Эйлеровым путем в графе называется путь, содержащий все ребра графа.
- Эйлеровым циклом или эйлеровой цепью называется цикл, содержащий все ребра графа и притом по одному разу.
- Граф, обладающий эйлеровым циклом, называется эйлеровым графом.
- Теорема. Связный граф называется эйлеровым тогда и только тогда, когда степени всех его вершин четны.

Алгоритм Флери – алгоритм поиска Эйлерова цикла

- Если все вершины чётны:

1) Произвольно выбирается некоторая вершина x_1 и ребро u_1 инцидентнтное x_1. Этому ребру присваивается номер 1. Вычеркиваем это ребро u_1 и переходим в вершину x_2 по ребру $u_1 = (x_1, x_2)$.

2) Находясь в вершине x_i, следует не выбирать ребро, соединяющее x_i с x_1, если имеется возможность иного выбора.

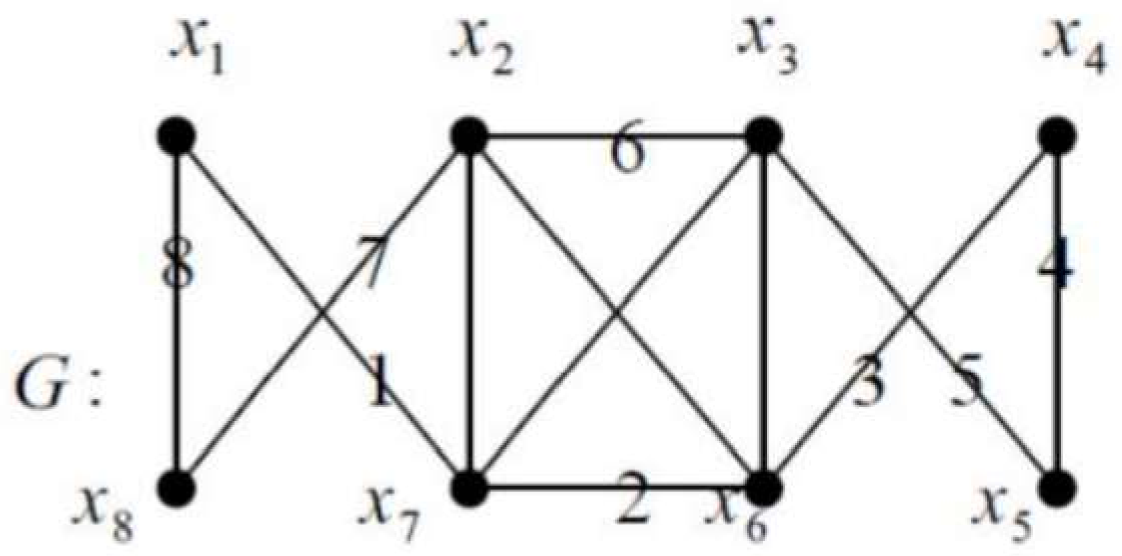
3) Находясь в вершине x_i, следует не выбирать ребро, которое является перешейком (т.е. ребром, при удалении которого, граф образованный вне вычеркнутыми ребрами, распадается на две компоненты связности, каждая из которых имеет хотя бы по одному ребру).

Алгоритм Флери – алгоритм поиска Эйлерова цикла

4) После того, как в графе будут занумерованы все ребра, образуется эйлеров цикл, причем порядок нумерации соответствует последовательности обхода ребер.

- Пример.

Построить эйлеров цикл в графе G.



Алгоритм Флери – алгоритм поиска Эйлерова цикла

Граф G – эйлеров, ибо $P(x_1) = P(x_4) = P(x_5) = P(x_8) = 2$, $P(x_2) = P(x_3) = P(x_6) = P(x_7) = 4$.

1) Выберем x_1 и ребро $u_1 = (x_1, x_7)$ и присвоим ему номер 1, затем перейдем в вершину $x_i = x_7$.

2) Находясь в x_7, не выбираем вычеркнутое ребро 1. Из оставшихся трех ребер ни одно не является перешейком, поэтому выбираем любое, например $u_2 = (x_7, x_6)$, присваиваем ему номер 2 и переходим в вершину $x_i = x_6$.

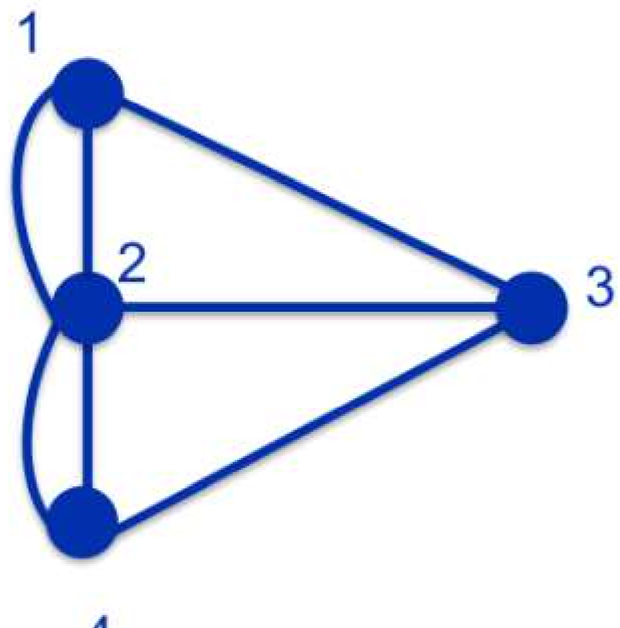
3) После восьми шагов опять приходим в вершину x_1. Построенный цикл $(x_1, x_7) - (x_7, x_6) - (x_6, x_4) - (x_4, x_5) - (x_5, x_3) - (x_3, x_2) - (x_2, x_8) - (x_8, x_1)$.

Определения

- Если граф содержит разомкнутую цепь, содержащую все ребра этого графа, то такой граф называется полуэйлеровым.
- Теорема 1. Если в связном графе две вершины нечетны, а все остальные – четны, то этот граф содержит эйлерову разомкнутую цепь.
- Теорема 2. Если в связном графе G содержится $2k$ нечетных вершин, то в нем имеется k разомкнутых эйлеровых цепей, в совокупности содержащих все ребра графа G точно по одному разу
- Теорема 3. В любом связном графе можно построить замкнутый маршрут, проходящий через каждое ребро точно два раза.

Эйлеровы графы

- Из теоремы 3 следует, что любой граф можно изобразить, не отрывая карандаш от бумаги и проходя по каждому ребру не более двух раз.
- Например, граф, приведенный ниже, можно изобразить в виде последовательности вершин так: 1, 2, 4, 2, 1, 3, 2, 3, 4, отсюда следует, что два раза карандаш прошел только по ребру {2,3}.



Гамильтоновы графы

- Гамильтоновым циклом, или путем в графе, называется цикл, или путь, проходящий через каждую вершину графа в точности по одному разу.
- Граф, обладающий гамильтоновым циклом, называется гамильтоновым графом.

Необходимые или достаточные условия гамильтоновости графа

- Если в графе есть висячая вершина (со степенью, равной единице), то гамильтонов цикл в нем отсутствует.
- Если граф на n вершинах является полным, то в нем имеется гамильтонов цикл при $n \geq 3$.
- Если в простом графе степень ρ каждой вершины удовлетворяет условию $\rho \geq n/2$, где $n \geq 3$, n – число вершин, то этот граф является гамильтоновым (теорема Дирака).

Необходимые или достаточные условия гамильтоновости графа

- Если для любой пары несмежных вершин v_i, v_j выполняется неравенство $\rho(v_i) + \rho(v_j) \geq n$, где $i, j, = 1,2, \dots, n; n \geq 3; i \neq j$; n – число вершин графа, то этот граф является гамильтоновым (теорема Оре)
- *Полный граф -простой неориентированный граф, в котором каждая пара различных вершин смежна, простой граф — граф, в котором нет кратных рёбер и петель.
- Связный граф, содержащий простую разомкнутую цепь, в которую входят все вершины графа, называется полугамильтоновым.

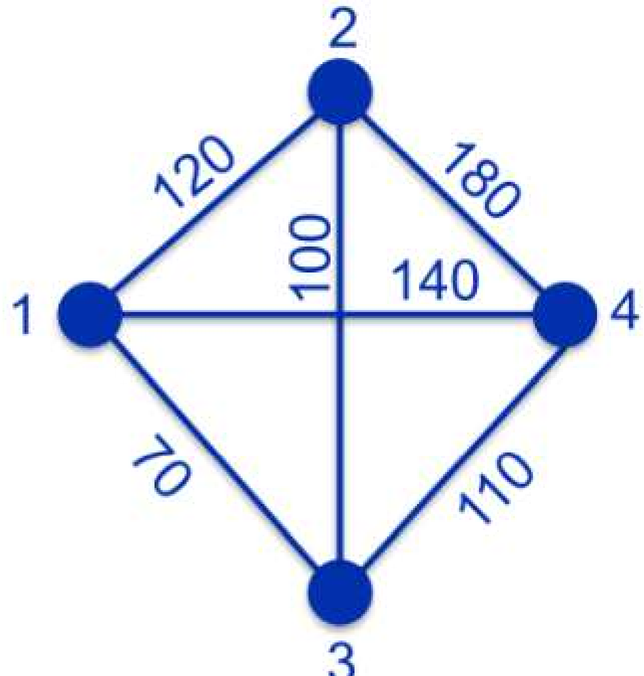
Задача о коммивояжере (о странствующем торговце)

- «Коммивояжер желает посетить n определенных городов; как он должен двигаться, чтобы заехать в каждый из них хотя бы один раз, проделав путь наименьшей общей длины?» (Коммивояжер может те или иные города посещать неоднократно.)
- «Коммивояжер обязан побывать в каждом пункте в точности по одному разу и заинтересован затратить на поездку как можно меньше времени».

Мы будем пользоваться второй формулировкой.

Задача о коммивояжере

- С математической точки зрения безразлично, какой параметр желает оптимизировать коммивояжер - время, расходы на дорогу или общую длину пути. В любом случае задача сводится к отысканию гамильтонова цикла.
- Рассмотрим граф:



Задача о коммивояжере

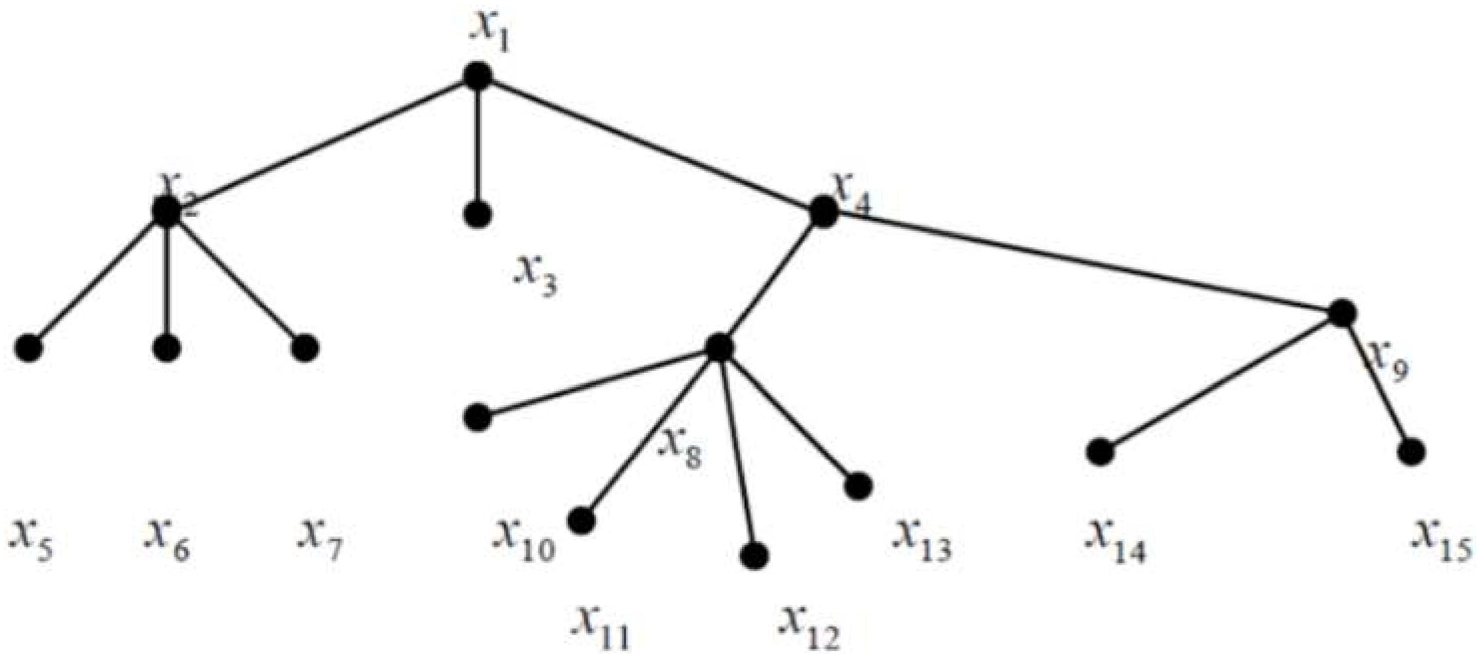
- В каком порядке коммивояжер должен обойти все города, преодолев наименьшее расстояние? В каком порядке он должен посетить города, если исходным является город 1?
- Методом отыскания всех простых цепей найдем все гамильтоновы циклы: а) 1, 2, 4, 3, 1; б) 1, 2, 3, 4, 1; в) 1, 3, 2, 4, 1; г) 1, 3, 4, 2, 1; д) 1, 4, 3, 2, 1; е) 1, 4, 2, 3, 1.
- Различны только: а, б, в. Длины путей:
  - а: 120 + 180 + 110 + 70 = 480;
  - б: 120 + 100 + 110 + 140 = 470;
  - в: 70 + 100 + 180 + 140 = 490.
- Таким образом, кратчайшим является путь б: 1, 2, 3, 4, 1.

Деревья. Алгоритм поиска минимальных остовных деревьев

- Деревом называется связный граф, не содержащий циклов.
- Любой граф без циклов называется лесом. Таким образом, деревья являются компонентами леса.
- Пусть $G = (S, U)$ и $|S| = n, |U| = m$. Тогда справедлива эквивалентность следующих утверждений:
  - G – дерево;
  - G – связный граф и $m = n - 1$;
  - G – ациклический граф и $m = n - 1$;
  - Любые две несовпадающие вершины графа соединяет единственная простая цепь;

Деревья. Алгоритм поиска минимальных остовных деревьев

- G – ациклический граф, обладающий тем свойством, что если какую-либо пару его несмежных вершин соединить ребром, то полученный граф будет содержать ровно один цикл.



Деревья. Алгоритм поиска минимальных остовных деревьев

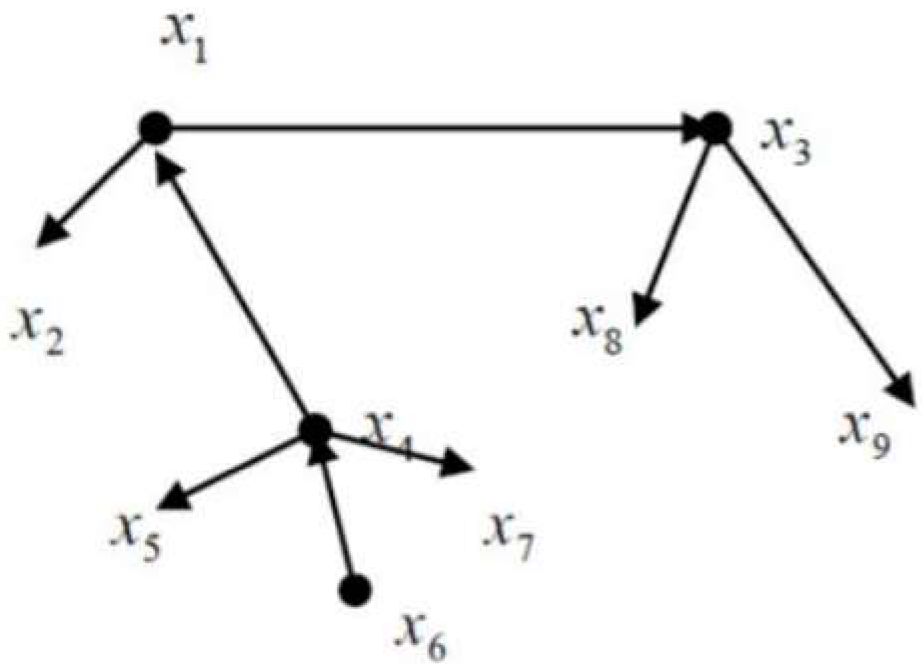
- Ориентированный граф называется ориентированным деревом (ордеревом), если:

1) существует ровна одна вершина $x_1 \in S$, называемая корнем, которая не имеет предшествующих вершин, т.е. $P(x_1) = 0$;

2) любой вершине $x_j \neq x_1$ в графе G непосредственно предшествует ровно одна вершина, т.е. $P(x_j) = 1$.

Деревья. Алгоритм поиска минимальных остовных деревьев

- Неориентированное дерево можно превратить в ориентированное, выбрав в качестве корня произвольную вершину.
- На рисунке корень графа - вершина x_6.



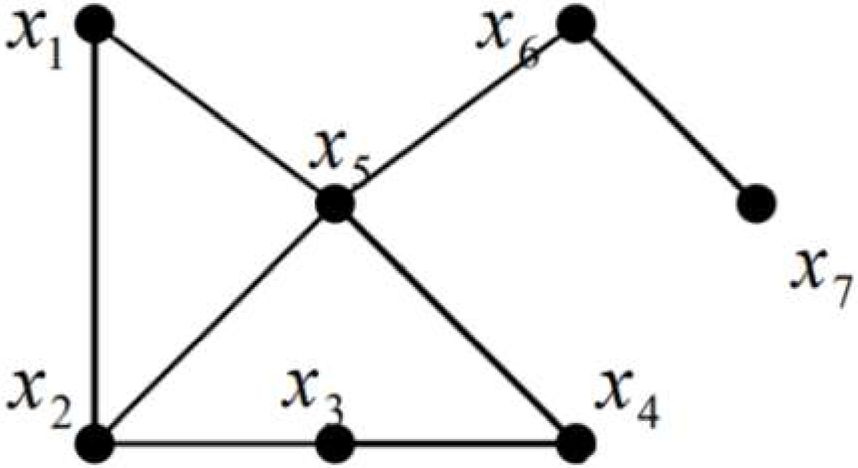
Деревья. Алгоритм поиска минимальных остовных деревьев

- Подграф $G' = (S', U)$ графа $G = (S, U)$ называется остовным подграфом, если $S' = S$.
- Подграф G' графа G называется остовным поддеревом (остовным каркасом), если $S' = S$ и G – дерево.
- Теорема Кэли. Число различных деревьев, которые можно построить на n различных вершинах, равно $t_n = n^{n-2}$.
- Теорема Кирхгофа. Число остовных деревьев в связном графе G порядка $n \geq 2$ равно алгебраическому дополнению любого элемента матрицы Кирхгофа $B(G)$.

Деревья. Алгоритм поиска минимальных остовных деревьев

- Теорема Кирхгофа. Пример.

  Подсчитаем по этой теореме число всех остовов графа



- Матрица Кирхгофа B определяется следующим образом:

$$b_{ij} = \begin{cases} -1, x_i \text{ и } x_j \text{ смежны}, \\ 0, x_i \text{ и } x_j \text{ несмежны и } i \neq j, \\ P(x_i), i = j \end{cases}$$

Деревья. Алгоритм поиска минимальных остовных деревьев

- Тогда

$$B = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{array} \begin{array}{c} \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{array} \\ \begin{bmatrix} 2 & -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{array}.$$

- Определим алгебраическое дополнение элемента b_{11}.

$$A_{11} = \begin{vmatrix} 3 & -1 & 0 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{vmatrix} = \begin{vmatrix} 3 & -1 & 0 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 4 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{vmatrix} + \begin{vmatrix} 3 & -1 & 0 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 4 & 0 \\ 0 & 0 & 0 & -1 & 2 \end{vmatrix} =$$

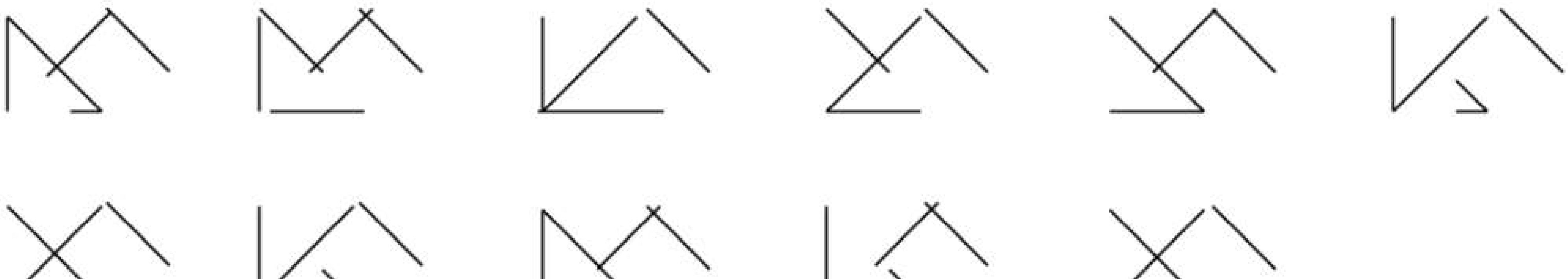
Деревья. Алгоритм поиска минимальных остовных деревьев

$$= -\begin{vmatrix} 3 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 4 \end{vmatrix} + \begin{vmatrix} 3 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & 0 \\ -1 & 0 & -1 & -1 \end{vmatrix} + 2 \cdot \begin{vmatrix} 3 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 4 \end{vmatrix} =$$

$$= 3 \cdot \begin{vmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{vmatrix} + \begin{vmatrix} -1 & 2 & -1 \\ 0 & -1 & 2 \\ -1 & 0 & -1 \end{vmatrix} + \begin{vmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ -1 & 0 & -1 \end{vmatrix} =$$

$$= 3(12 + 0 + 0 - 0 - 2 - 3) + (-1 - 4 + 0 + 1 - 0 - 0) + (-6 - 1 + 0 - 0 + 1 - 1) = 11$$

- Таким образом у этого графа существует 11 различных остовов:



Деревья. Алгоритм поиска минимальных остовных деревьев

- Теорема. Число ребер произвольного неориентированного графа G, которые необходимо удалить для получения остова, не зависит от последовательности их удаления и равно $v(G) = m - n + k$, где m – число ребер, n – число вершин и k – число компонент связности графа G.

Следствие 1. Неориентированный граф G является лесом тогда и только тогда, когда $v(G) = 0$.

Следствие 2. Неориентированный граф G имеет единственный цикл тогда и только тогда, когда $v(G) = 1$.

Задача об остове экстремального веса

- Пусть $G = (S, U)$ – связная сеть.
- Пусть $G = (S, U, \Omega)$ служит моделью железнодорожной сети, соединяющей пункты $x_1, x_2, \dots, x_n \in S$, а $\omega(x_i, x_j)$ - расстояние между пунктами x_i и x_j.
- Требуется проложить сеть телеграфных линий вдоль железнодорожной сети так, чтобы все пункты $x_1, x_2, \dots, x_n$ были связаны между собой телеграфной сетью и общая протяженность линий телеграфной сети была наименьшей.

Задача об остове экстремального веса

- Алгоритм Прима (алгоритм ближайшего соседа):

состоит из двух шагов и выполняется $n-1$ раз на графе G с n вершинами.

- Пусть $S' \subset S, S'' \subset S$ и $S = S' \cup S'', S' \cap S'' = \emptyset$.

Определим пошаговое расстояние между множествами S' и S'':

$$d(S', S'') = min\{\omega(x_i, x_j) | x_i \in S', x_j \in S''\}$$

(x_i, x_j) – дуга, соединяющая вершины x_i и x_j.

Задача об остове экстремального веса

- Основные шаги алгоритма:

1) Присвоение начальных значений.

Полагают $S' = \{x_1\}$, где x_1 - произвольная вершина, $S'' = S \setminus S', U' = \emptyset$.

2) Обновление данных.

Находится ребро (x_i, x_j) такое, что $x_i \in S'$, $x_j \in S''$

$$\omega(x_i, x_j) = min\{\omega(x_i, x_j) | x_i \in S', x_j \in S''\}$$

Полагают $S' = S' \cup \{x_j\}, S'' = S \setminus S', U' = U' \cup \{(x_i, x_j)\}$.

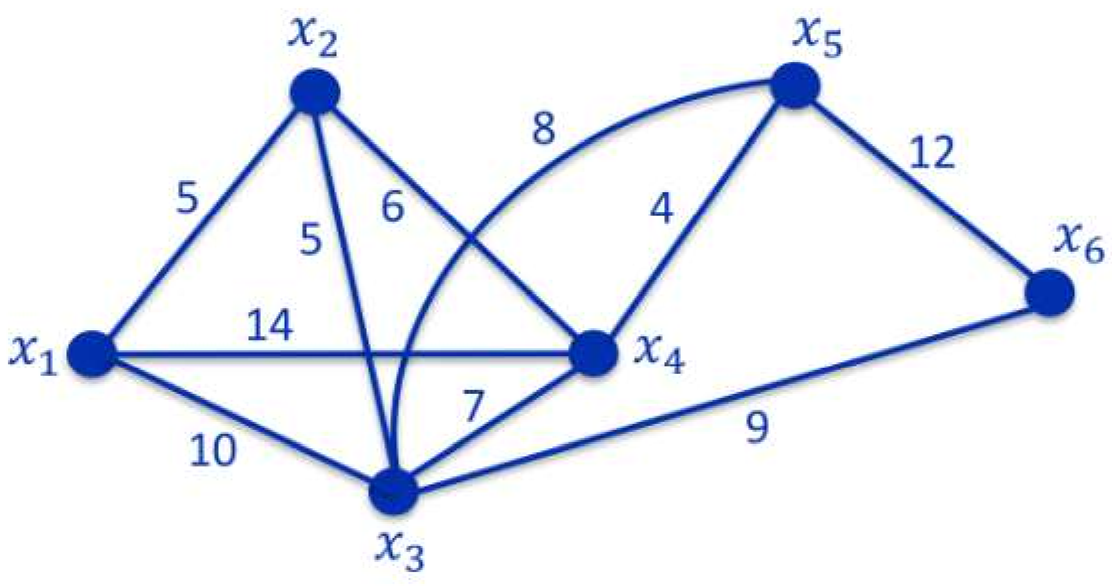
3) Проверка на завершение.

Если $S' = S$, то $G' = (S', U')$ – искомый остов. В противном случае переходят ко второму шагу.

Задача об остове экстремального веса

<u>Пример</u>. Построить остов с наименьшим весом для сети, заданной матрицей весом Ω.

$$\Omega = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{bmatrix} 0 & 5 & 10 & 14 & 0 & 0 \\ 5 & 0 & 5 & 6 & 0 & 0 \\ 10 & 5 & 0 & 7 & 8 & 9 \\ 14 & 6 & 7 & 0 & 4 & 0 \\ 0 & 0 & 8 & 4 & 0 & 12 \\ 0 & 0 & 9 & 0 & 12 & 0 \end{bmatrix} \end{array}$$



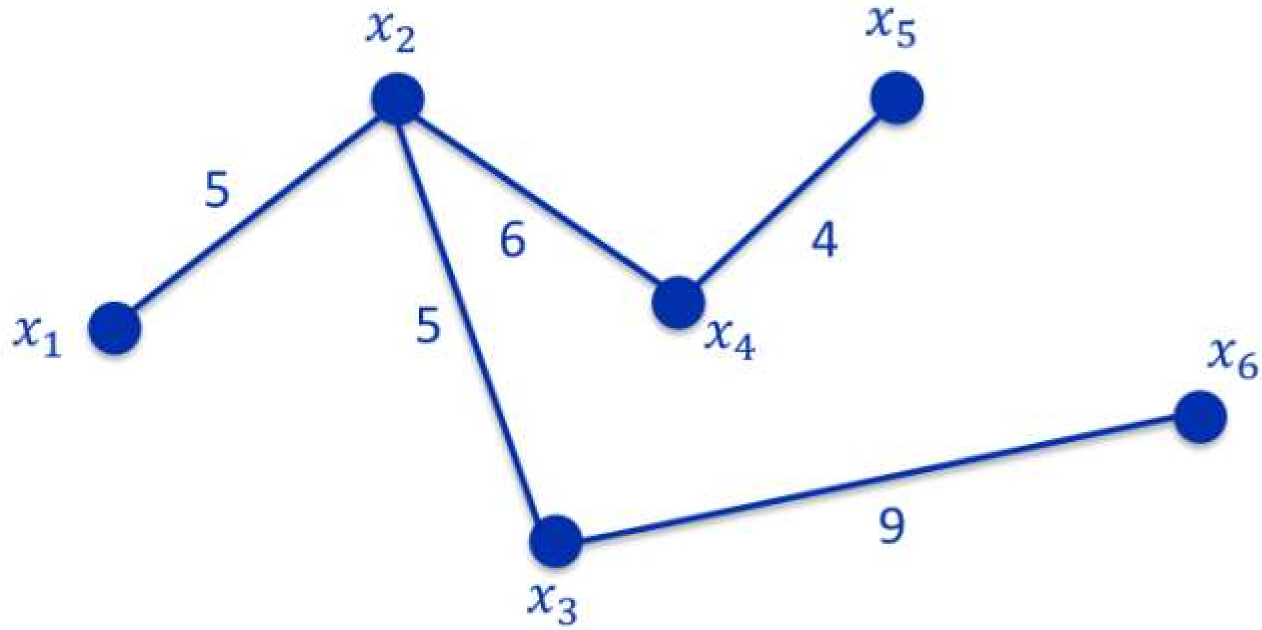
Задача об остове экстремального веса

- $S' = \{x_1\}, S'' = \{x_2, x_3, x_4, x_5, x_6\}, U' = \emptyset.$

Итерация	$d(S', S'')$	S'	S''	U'
1	$\omega(x_1, x_2) = 5$	$\{x_1, x_2\}$	$\{x_3, x_4, x_5, x_6\}$	$\{(x_1, x_2)\}$
2	$\omega(x_2, x_3) = 5$	$\{x_1, x_2, x_3\}$	$\{x_4, x_5, x_6\}$	$\{(x_1, x_2), (x_2, x_3)\}$
3	$\omega(x_2, x_4) = 6$	$\{x_1, x_2, x_3, x_4\}$	$\{x_5, x_6\}$	$\{(x_1, x_2), (x_2, x_3), (x_2, x_4)\}$
4	$\omega(x_4, x_5) = 4$	$\{x_1, x_2, x_3, x_4, x_5\}$	$\{x_6\}$	$\{(x_1, x_2), (x_2, x_3), (x_2, x_4), (x_4, x_5)\}$
5	$\omega(x_3, x_6) = 9$	$\{x_1, x_2, x_3, x_4, x_5, x_6\}$	$\emptyset$	$\{(x_1, x_2), (x_2, x_3), (x_2, x_4), (x_4, x_5), (x_3, x_6)\}$

Задача об остове экстремального веса

- S'=S. Получен остовный граф.
- $G' = (S', U')$, вес $\omega(G') = 5 + 5 + 6 + 4 + 9 = 29$.



УНИВЕРСИТЕТ ИТМО

Спасибо за внимание!

Санкт-Петербург, 2022