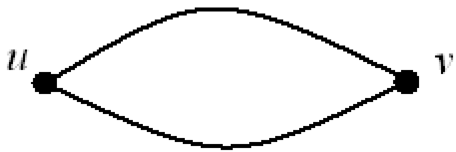
§7. Деревья

Опр. Деревом называется обыкновенный связный граф без циклов.

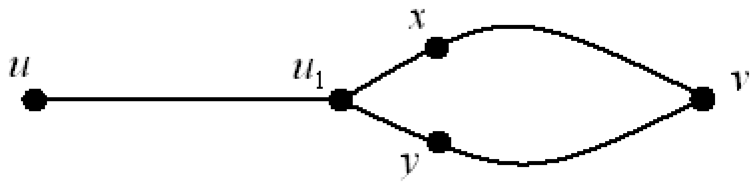
Замечание: в дереве любое ребро является мостом.

Теорема (о деревьях).
Пусть G – обыкновенный граф. Тогда следующие условия эквивалентны:
(1) G – дерево.
(2) Любые две различные вершины соединены единственной простой цепью.
(3) G не содержит циклов. Но если любую пару несмежных вершин соединить ребром, возникает в точности один простой цикл.

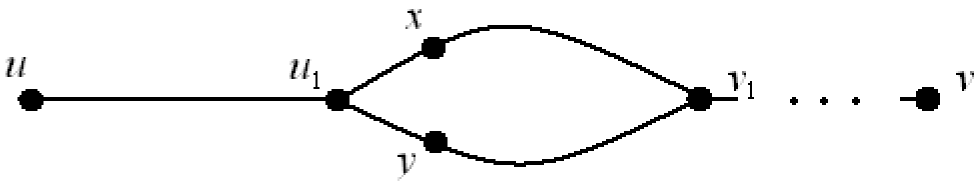
Доказательство:
(1) $\Rightarrow$ (2) Дано: G – обыкновенный связный граф без циклов. Следовательно, любые две вершины соединены цепью, а значит и простой цепью.
(от противного) Предположим, существуют вершины u и v, соединенные двумя различными простыми цепями. Если все вершины в этих цепях различны, получается цикл – противоречие.



Предположим, что некоторые вершины в цепях совпадают. Пусть *x* и *y* – ближайшие к *u* вершины, в которых цепи не совпадают, тогда предыдущая вершина u_1 соединяется с *v* несовпадающими цепями. Если все вершины в этих цепях различны, получается цикл – противоречие.



Если в этих цепях есть общие вершины, выберем из них v_1 – ближайшую к x и y. Тогда вершина u_1 соединяется с v_1 несовпадающими цепями, в которых все вершины различны, получается цикл – противоречие.

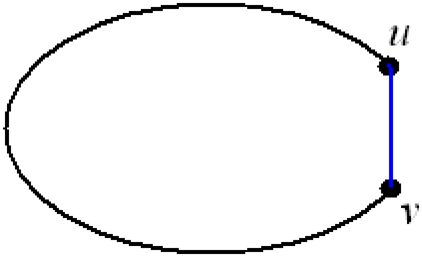


Следовательно, не существуют вершины u и v, соединенные двумя различными простыми цепями.

$(2) \Rightarrow (3)$ Дано: любые две различные вершины в G соединены единственной простой цепью.

Если в G существует цикл (простой), то найдутся две вершины, соединённые двумя различными цепями – противоречие. Следовательно, в G нет циклов.

Пусть две различные вершины u и v не смежны. Постоим граф $G_1 = G \cup (u, v)$. Тогда в G_1 найдётся простой цикл, соединяющий u и v.



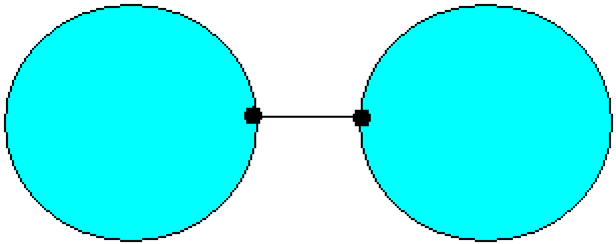
Предположим, что в G_1 существуют два различных простых цикла C_1 и C_2. Очевидно, они оба содержат ребро (u, v). Т.е. отличаются ребрами из $C_1 \setminus (u, v)$ и $C_2 \setminus (u, v)$.

Тогда $C_1 \setminus (u, v)$ и $C_2 \setminus (u, v)$ – несовпадающие простые (u–v)-цепи. Противоречие.

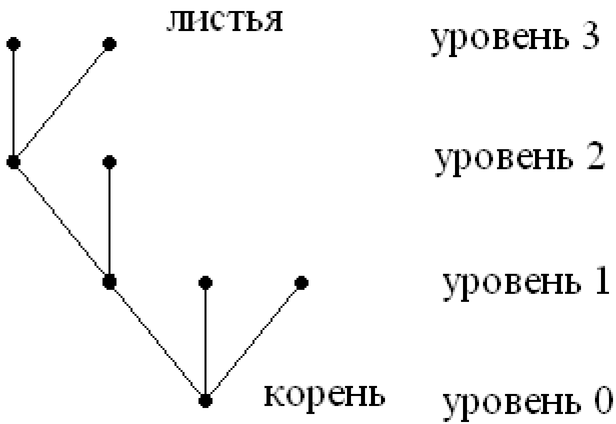
Следовательно цикл в G_1 только один.

(3) $\Rightarrow$ (1) Дано: G не содержит циклов. Но если любую пару несмежных вершин соединить ребром, возникает в точности один простой цикл.

Остаётся доказать, что G связный.
(от противного) Предположим, что существуют вершины u и v, не соединённые цепью, т.е. из разных компонент связности. Тогда в графе $G_1 = G \cup (u, v)$ не может быть цикла, проходящего через ребро (u, v). Противоречие. Теорема доказана.

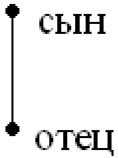


Стандартное изображение дерева.
1 вариант:

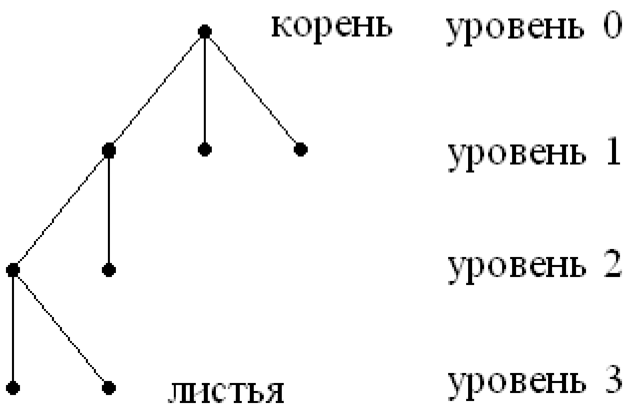


Пример. Генеалогическое дерево потомков по мужской линии.

Следствие:



2 вариант:



Пример. Иерархия подразделений фирмы (УрФУ).

Теорема.
Любое дерево допускает стандартное изображение.

Доказательство:
Алгоритм построения стандартного изображения.

1. Выбрать любую вершину v_0 – корень дерева.

2. Собрать множество $V_1 = \{v \mid (v_0, v) \in E\}$. Это уровень 1.

3. Собрать множество $V_2 = \{v \mid \exists u \in V_1 : (u, v) \in E\} \setminus \{v_0\}$. Это уровень 2.

4. Собрать множество $V_3 = \{v \mid \exists u \in V_2 : (u,v) \in E\} \setminus V_1$. Это уровень 3.

…

k + 1. Собрать множество $V_k = \{v \mid \exists u \in V_{k-1} : (u,v) \in E\} \setminus V_{k-2}$. Это уровень k.

Алгоритм останавливается, когда $V_k = \varnothing$.

Т.к. G дерево, одна вершина не может принадлежать различным уровням. Следовательно, $\{v_0\}, V_1, V_2, \ldots, V_k$ является разбиением множества вершин дерева. Ребра соединяют только вершины соседних уровней.

Следствие.
Если G – дерево, то $m = n - 1$, где m – количество ребер, n – количество вершин.

Доказательство:
В стандартном изображении каждая вершина, кроме корня, имеет единственного отца. Количество ребер равно количеству вершин, от которых идет ребро к отцу. Следствие доказано.

Опр. Лесом называется обыкновенный несвязный граф без циклов.

§8. Цикломатическое число. Каркас.

Опр. Цикломатическим числом графа G называется $\gamma(G) = m - n + c$, где m – число ребер, n – число вершин, c – число компонент связности G.

Пример. Если G – дерево, то $\gamma(G) = (n-1) - n + 1 = 0$.

Теорема.

Если граф G состоит из компонент связности $G_1, G_2, \ldots, G_c$, то $\gamma(G) = \gamma(G_1) + \gamma(G_2) + \ldots + \gamma(G_c)$.

Доказательство:

Для каждого i выполняется $\gamma(G_i) = m_i - n_i + 1$.

$$\sum_{i=1}^{c} m_i = m, \ \sum_{i=1}^{c} n_i = n.$$

$$\Rightarrow \sum_{i=1}^{c} \gamma(G_i) = \sum_{i=1}^{c} m_i - \sum_{i=1}^{c} n_i + \sum_{i=1}^{c} 1 = m - n + c.$$

Пример.

Если G – лес, то $\gamma(G) = 0$.

Теорема.
(1) Из любого связного графа G можно удалить $\gamma(G)$ ребер, так, что оставшийся суграф – дерево.

(2) При удалении менее $\gamma(G)$ ребер в суграфе останется цикл; при удалении более $\gamma(G)$ ребер суграф становится несвязным.

Доказательство:
(1) Граф G связный.
Если в нём нет циклов, то G – дерево, следовательно, $\gamma(G) = 0$.

Если в G есть цикл, то удалив любое ребро e, принадлежащее циклу, получим связный граф $G_1 = G \setminus e$.

$$\gamma(G_1) = (m-1) - n + 1 = \gamma(G) - 1.$$

Повторим удаление ребер, принадлежащих циклу, пока не получим связный граф без циклов (дерево) G_k.
$$\gamma(G_k) = \gamma(G) - k = 0 \Rightarrow k = \gamma(G).$$

(2) Граф G связный.

Если G – дерево, то $\gamma(G) = 0$. Менее 0 ребер удалить невозможно.

Удалив хотя бы одно ребро в дереве (мост), получим несвязный граф.

Если G – не дерево, то в G есть цикл.

Удалив ребро, принадлежащее циклу, получим граф G_1, где

$\gamma(G_1) = \gamma(G) - 1$.

Удалив ребро, не принадлежащее циклу, получим граф G_2, где

$\gamma(G_2) = (m-1) - n + (c+1) = \gamma(G)$.

Следовательно, удалив любые s ребер, получим граф G^*, где $\gamma(G^*) \geq \gamma(G) - s$.

Если $s < \gamma(G)$, то $\gamma(G^*) > 0$. G^* не является ни деревом, ни лесом, т.е. содержит цикл.

Если $s > \gamma(G)$, то среди удаляемых ребер найдется хотя бы один мост, следовательно, G^* не связный.

Следствие.
Если в связном графе G выполняется $\gamma(G)=0$, то G – дерево.

Опр. Каркасом связного графа G называется суграф T, являющийся деревом (полученный удалением $\gamma(G)$ подходящих ребер).

Опр. Каркасом несвязного графа G называется объединение каркасов компонент связности, т.е. лес.

§9. Задача о минимальном соединении. «Жадные» алгоритмы

Одна из оптимизационных задач на графах.

Пусть $G = (V, E)$ – связный обыкновенный граф, на ребрах которого задана функция $\mu : E \to R^+$.

Назовём $\mu(e)$ весом ребра e (или стоимостью ребра), а функцию μ – весовой функцией.

Задача о минимальном соединении:
Для заданных графа G и весовой функции μ найти каркас T с наименьшим суммарным весом ребер, т.е.

$$\mu(T) = \sum_{e \in T} \mu(e) \to \min$$

Замечание: такой каркас всегда существует, т.к. из всех каркасов всегда можно выбрать каркас с наименьшим суммарным весом ребер.

Требуется привести алгоритм, не использующий перебор всех каркасов.

Алгоритм Краскала.

1. Выбрать в E ребро e_1 с наименьшим весом $\mu(e_1)$ (любое среди ребер с одинаковым маленьким весом).
Собрать множество $T_1 = \{e_1\}$.

Собрать $E_1 = \{e \mid e \notin T_1, (T_1 \cup e)$ не содержит циклов $\}$
Если $E_1 = \varnothing$, то алгоритм заканчивает работу.

2. Выбрать в E_1 ребро e_2 с наименьшим весом $\mu(e_2)$.
Собрать множество $T_2 = T_1 \cup e_2$.
Собрать $E_2 = \{e \mid e \notin T_2, (T_2 \cup e)$ не содержит циклов $\}$.
Если $E_2 = \varnothing$, то алгоритм заканчивает работу.

…

k. Выбрать в E_{k-1} ребро e_k с наименьшим весом $\mu(e_k)$.

Собрать множество $T_k = T_{k-1} \cup e_k$.

Собрать $E_k = \{e \mid e \notin T_k, (T_k \cup e) \text{ не содержит циклов} \}$.

Если $E_k = \varnothing$, то алгоритм заканчивает работу.

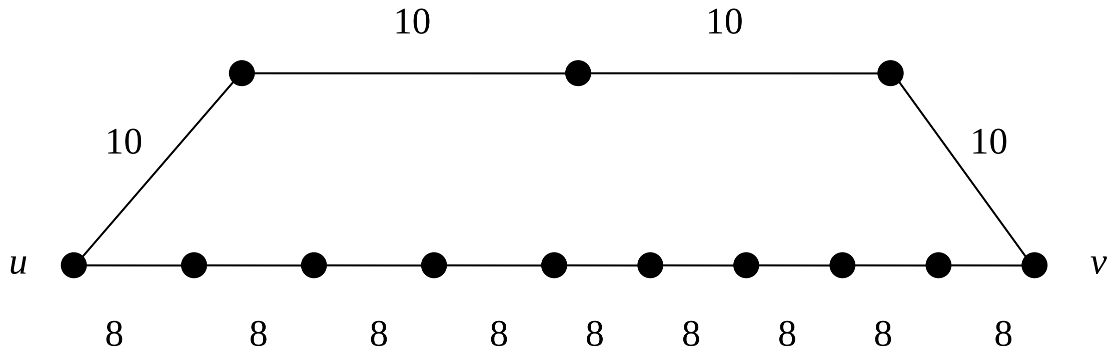
…

Множество T_k, полученное после остановки алгоритма, является искомым каркасом, причем количество ребер в T_k будет равно $n - 1$.

Для улучшения работы алгоритма можно использовать множество E_k, упорядоченное по возрастанию весов, тогда на каждом шаге выбираем первое ребро в списке.

Замечание: Алгоритм реализует «жадную» стратегию, на каждом шаге выбирая наименьшее по стоимости ребро, так, чтобы не получались циклы. Существуют задачи, где «жадная» стратегия не даёт оптимальный результат.

Пример. Найти маршрут наименьшей стоимости между парой вершин u и v в графе на рисунке.



Теорема.
Для любого связного графа G и любой весовой функции μ алгоритм Краскала даёт оптимальное решение задачи о минимальном соединении.

Доказательство: Пусть T_k – граф, полученный алгоритмом Краскала.
1. Покажем, что T_k суграф графа G, т.е. T_k содержит ребра, инцидентные всем вершинам в G.
(от противного) Предположим, что вершина $v \notin T_k$. В G вершина v не изолированная, значит, существует ребро e, инцидентное v, и $e \notin T_k$.
Тогда $T_k \cup e$ не содержит циклов, т.е. $E_k \neq \varnothing$. Противоречие.

2. Покажем, что граф T_k связный.

(от противного) Предположим, что граф T_k несвязный, однако G связный. Тогда существует ребро (u, v), где u и v лежат в разных компонентах связности в T_k. Граф $T_k \cup (u,v)$ не содержит циклов, и $E_k \neq \varnothing$. Противоречие.

3. Покажем, что $\mu(T_k)$ наименьшая.

(от противного) Предположим, что существует каркас T, в котором $\mu(T) < \mu(T_k)$.

Т.к. $\mu(T) < \mu(T_k)$, существует ребро $e \in T_k$, $e \notin T$ ($e \in T_k \setminus T$).

Пусть T – каркас графа G с наименьшей $\mu(T)$ и наименьшим количеством ребер в $T_k \setminus T$.

Рассмотрим ребро $e_j \in T_k \setminus T$, добавленное на шаге j с наименьшим номером (самое первое ребро, не принадлежащее T)

Граф $T \cup e_j$ содержит единственный простой цикл C. В этом цикле существует ребро $e' \in C$, $e' \notin T_k$.

Построим граф $T' = (T \cup e_j) \setminus e'$, связный без циклов.

$\mu(T) \leq \mu(T') \Rightarrow \mu(e') \leq \mu(e_j)$.

Предположение $\mu(e') < \mu(e_j)$ противоречит выбору e_j, следовательно $\mu(e') = \mu(e_j)$.

Т.о. $\mu(T) = \mu(T')$, но количество ребер в $T_k \setminus T'$ меньше, чем в $T_k \setminus T$. Противоречие.

Замечание. Существует описание алгебраических структур, в которых «жадные» алгоритмы дают оптимальное решение оптимизационных задач. Такие структуры называют матроидами.