

Лекция №13

01.11.2024

2. Задача о путях (продолжение)

Теорема. Матрица стоимостей размеченного над полукольцом орграфа равна итерации его матрицы меток дуг: $C = A^*$.

Доказательство. Нам потребуется понятие ограниченной стоимости прохождения.

Пусть дано некоторое множество путей \mathbf{W} орграфа, размеченного над некоторым полукольцом. Например, это может быть множество путей заданной длины.

Тогда стоимость прохождения из i -й вершины в j -ю, ограниченная на множество путей \mathbf{W} есть, по определению

$$c_{ij} |_{\mathbf{W}} = \begin{cases} 0, & \text{если нет ни одного пути из } i\text{-й вершины в } j\text{-ю, принадлежащего множеству } \mathbf{W} \\ \sum_{\substack{W: v_i \Rightarrow^* v_j, \\ W \in \mathbf{W}}} \varphi^*(W) & \end{cases}$$

Сумма меток берется по всем путям, принадлежащим множеству \mathbf{W} . Если оно совпадает с множеством всех путей графа, получаем определенную выше общую стоимость прохождения.

Лемма. Матрица $A^m, m \geq 0$, (m -я степень матрицы меток дуг) равна матрице стоимостей по всем путям длины m .

Доказательство леммы. Индукция по m . При $m = 0$ получаем единичную матрицу, которая, очевидно, есть матрица стоимостей по всем путям длины 0.

Пусть для всех $m \leq l-1, l > 0$, матрица A^m есть матрица стоимостей по всем путям длины m .

Рассмотрим тогда матрицу $A^l = A^{l-1} A$. Обозначая элемент матрицы A^m как $a_{ij}^{(m)}, 0 \leq m \leq l$, получим:

$$a_{ij}^{(l)} = \sum_{k=1}^n a_{ik}^{(l-1)} a_{kj} = \begin{cases} 0, & \text{если } (\forall k)(\neg(v_i \Rightarrow^{*_{l-1}} v_k) \vee a_{kj} = 0) \\ \sum_{k=1}^n \left(\sum_{W: v_i \Rightarrow^* v_k, |W|=l-1} \varphi^*(W) \right) a_{kj} = \sum_{k=1}^n \left(\sum_{W: v_i \Rightarrow^* v_k, |W|=l-1} \varphi^*(W) a_{kj} \right), & \text{иначе} \end{cases}$$

То есть элемент $a_{ij}^{(l)}$ равен нулю полукольца, если либо нет путей длины $l-1$, ведущих из i -й вершины в k -ю (какую-то промежуточную на пути из i -й вершины в j -ю), либо нет дуги, ведущей из k -й вершины в j -ю, какова бы ни была при этом промежуточная вершина. Это означает, что вообще нельзя пройти из i -й вершины в j -ю по пути длины l .

Иначе, то есть, если такая промежуточная k -я вершина найдется, что в нее из i -й вершины можно пройти по пути длины $l-1$, а из нее по дуге попасть в j -ю вершину, то, используя предположение индукции, заменяем $a_{ik}^{(l-1)} = \sum_{W: v_i \Rightarrow^* v_k, |W|=l-1} \varphi^*(W)$ (внутренняя

сумма выше), что является стоимостью прохождения из i -й вершины в k -ю по всем путям длины $l-1$. Эта сумма конечная, так как множество путей фиксированной длины конечно. Используя дистрибутивность умножения относительно сложения, вносим множитель a_{kj} (метку дуги из k -й вершины в j -ю) под указанную конечную сумму справа, в результате чего получаем, во внутренней сумме, стоимость прохождения из i -й вершины в j -ю по таким путям длины l , которые оканчиваются дугой $v_k \rightarrow v_j$ при фиксированном k , номере промежуточной вершины. Меняя теперь промежуточную вершину произвольно, то есть делая ее любой вершиной графа, получим полную стоимость прохождения из i -й вершины в j -ю по всем возможным путям длины l .

$$v_i \Rightarrow a_{ik}^{(l-1)} \Rightarrow v_k \xrightarrow{a_{kj}} v_j$$

Лемма доказана.

С учетом доказанной леммы тогда итерация матрицы меток дуг

$$A^* = \sum_{m=0}^{\infty} A^m$$

выражает стоимости прохождения по путям произвольной длины, то есть оказывается матрицей стоимостей размеченного орграфа.

Теорема доказана.

Алгоритм вычисления итерации матрицы

Матрица стоимостей может быть найдена как решение матричного уравнения

$$X = AX + E,$$

где E - единичная матрица.

Действительно, решение есть $X = A^* E = A^*$.

Практически матрица стоимостей вычисляется по столбцам, а именно, очередной j -й ее столбец находится как решение системы линейных уравнений

$$\xi_j = A\xi_j + \varepsilon_j, 1 \leq j \leq n,$$

где ε_j - j -й столбец единичной матрицы.

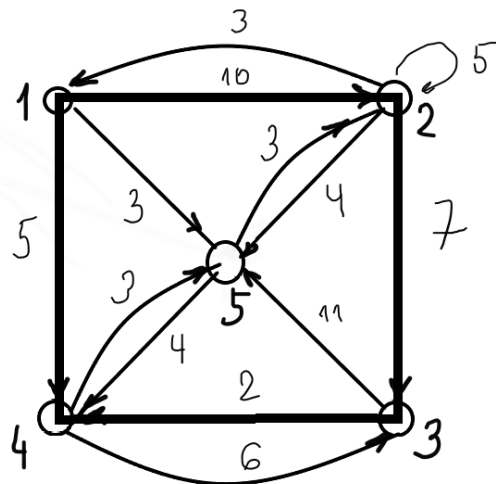
Другими словами, относительно компонент неизвестного j -го столбца искомой матрицы записывается система линейных уравнений в полукольце разметки и в j -м уравнении (и только в нем!) добавляется справа свободный член в виде единицы полукольца:

$$\begin{cases} x_1 = a_{11}x_1 + \dots + a_{1j}x_j + \dots + a_{1n}x_n \\ \vdots \\ x_j = a_{j1}x_1 + \dots + a_{jj}x_j + \dots + a_{jn}x_n + 1 \\ \vdots \\ x_n = a_{n1}x_1 + \dots + a_{nj}x_j + \dots + a_{nn}x_n \end{cases}$$

$1 \leq j \leq n.$

Решение примера

Для графа, заданного матрицей меток дуг, найти матрицу кратчайших расстояний:



$$A = \begin{pmatrix} +\infty & 10 & +\infty & 5 & 3 \\ 3 & 5 & 7 & +\infty & 4 \\ +\infty & +\infty & +\infty & 2 & 11 \\ +\infty & +\infty & 6 & +\infty & 3 \\ +\infty & 3 & +\infty & 4 & +\infty \end{pmatrix}$$

Вычисляем 1-й столбец:

$$\begin{cases} x_1 = 10 \cdot x_2 + 5 \cdot x_4 + 3 \cdot x_5 + 0 \Rightarrow x_1 = 0 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 7 \cdot x_3 + 4 \cdot x_5 \\ x_3 = 2 \cdot x_4 + 11 \cdot x_5 \\ x_4 = 6 \cdot x_3 + 3 \cdot x_5 \\ x_5 = 3 \cdot x_2 + 4 \cdot x_4 \end{cases}$$

Далее:

$$\begin{cases} x_2 = 5 \cdot x_2 + 7 \cdot x_3 + 4 \cdot x_5 + 3 \Rightarrow x_2 = 3 \\ x_3 = 2 \cdot x_4 + 11 \cdot x_5 \\ x_4 = 6 \cdot x_3 + 3 \cdot x_5 \\ x_5 = 3 \cdot x_2 + 4 \cdot x_4 \end{cases}$$

$$\begin{cases} x_3 = 2 \cdot x_4 + 11 \cdot x_5 \\ x_4 = 6 \cdot x_3 + 3 \cdot x_5 \\ x_5 = 4 \cdot x_4 + 6 \end{cases}$$

$$\begin{cases} x_3 = 2 \cdot x_4 + 11 \cdot (4 \cdot x_4 + 6) \\ x_4 = 6 \cdot x_3 + 3 \cdot (4 \cdot x_4 + 6) \end{cases}$$

$$\begin{cases} x_3 = 2 \cdot x_4 + 17 \\ x_4 = 6 \cdot x_3 + 7 \cdot x_4 + 9 \end{cases}$$

$$\{x_4 = 6 \cdot (2 \cdot x_4 + 17) + 7 \cdot x_4 + 9 = 7 \cdot x_4 + 9 \Rightarrow x_4 = 9$$

$$x_3 = 2 \cdot 9 + 17 = 11;$$

$$x_5 = 4 \cdot 9 + 6 = 6.$$

Итак, 1-й столбец

$$\xi_1 = (0, 3, 11, 9, 6)^T.$$

Как видно, нетривиальных сокращений расстояний (относительно дуг) не произошло, но исчезли все бесконечности.

2-й столбец

$$\begin{cases} x_1 = 10 \cdot x_2 + 5 \cdot x_4 + 3 \cdot x_5 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 7 \cdot x_3 + 4 \cdot x_5 + 0 \Rightarrow x_2 = 0 \\ x_3 = 2 \cdot x_4 + 11 \cdot x_5 \\ x_4 = 6 \cdot x_3 + 3 \cdot x_5 \\ x_5 = 3 \cdot x_2 + 4 \cdot x_4 \end{cases}$$

$$\begin{cases} x_1 = 5 \cdot x_4 + 3 \cdot x_5 + 10 \\ x_3 = 2 \cdot x_4 + 11 \cdot x_5 \\ x_4 = 6 \cdot x_3 + 3 \cdot x_5 \\ x_5 = 4 \cdot x_4 + 3 \Rightarrow x_5 = 3 \end{cases}$$

$$\begin{cases} x_1 = 5 \cdot x_4 + 6 \Rightarrow x_1 = 6 \\ x_3 = 2 \cdot x_4 + 14 \Rightarrow x_3 = 8 \\ x_4 = 6 \cdot x_3 + 6 \Rightarrow x_4 = 6 \end{cases}$$

Итак, 2-й столбец:

$$\xi_1 = (6, 0, 8, 6, 3)^T$$

Здесь можно отметить сокращение метки пути из 1 вершины во вторую (через пятую) по сравнению с дугой: $6 < 10$.

3-й столбец

$$\begin{cases} x_1 = 10 \cdot x_2 + 5 \cdot x_4 + 3 \cdot x_5 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 7 \cdot x_3 + 4 \cdot x_5 \\ x_3 = 2 \cdot x_4 + 11 \cdot x_5 + 0 \Rightarrow x_3 = 0 \\ x_4 = 6 \cdot x_3 + 3 \cdot x_5 \\ x_5 = 3 \cdot x_2 + 4 \cdot x_4 \end{cases}$$

$$\begin{cases} x_1 = 10 \cdot x_2 + 5 \cdot x_4 + 3 \cdot x_5 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 4 \cdot x_5 + 7 \\ x_4 = 3 \cdot x_5 + 6 \\ x_5 = 3 \cdot x_2 + 4 \cdot x_4 \end{cases}$$

Здесь нет возможности использовать принцип минимума, поэтому исключаем x_4 (проще всего):

$$\begin{cases} x_1 = 10 \cdot x_2 + 5 \cdot (3 \cdot x_5 + 6) + 3 \cdot x_5 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 4 \cdot x_5 + 7 \\ x_5 = 3 \cdot x_2 + 4 \cdot (3 \cdot x_5 + 6) \end{cases}$$
$$\begin{cases} x_1 = 10 \cdot x_2 + 3 \cdot x_5 + 11 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 4 \cdot x_5 + 7 \\ x_5 = 3 \cdot x_2 + 7 \cdot x_5 + 10 \Rightarrow x_5 = 3 \cdot x_2 + 10 \end{cases}$$

$$\begin{cases} x_1 = 10 \cdot x_2 + 3 \cdot (3 \cdot x_2 + 10) + 11 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 4 \cdot (3 \cdot x_2 + 10) + 7 \end{cases}$$

$$\begin{cases} x_1 = 6 \cdot x_2 + 11 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 7 \end{cases}$$

$$\{x_2 = 3 \cdot (6 \cdot x_2 + 11) + 5 \cdot x_2 + 7$$

$$x_2 = 5 \cdot x_2 + 7 \Rightarrow x_2 = 7; x_5 = 10, x_4 = 6, x_1 = 11$$

3-й столбец:

$$\xi_3 = (11, 7, 0, 6, 10)^T$$

4-й столбец

$$\begin{cases} x_1 = 10 \cdot x_2 + 5 \cdot x_4 + 3 \cdot x_5 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 7 \cdot x_3 + 4 \cdot x_5 \\ x_3 = 2 \cdot x_4 + 11 \cdot x_5 \\ x_4 = 6 \cdot x_3 + 3 \cdot x_5 + 0 \Rightarrow x_4 = 0 \\ x_5 = 3 \cdot x_2 + 4 \cdot x_4 \end{cases}$$

$$\begin{cases} x_1 = 10 \cdot x_2 + 3 \cdot x_5 + 5 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 7 \cdot x_3 + 4 \cdot x_5 \\ x_3 = 11 \cdot x_5 + 2 \Rightarrow x_3 = 2 \\ x_5 = 3 \cdot x_2 + 4 \Rightarrow x_5 = 4 \end{cases}$$

$$\begin{cases} x_1 = 10 \cdot x_2 + 5 \Rightarrow x_1 = 5 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 7 \cdot x_3 + 4 \cdot x_5 = 3 \cdot 5 + 7 \cdot 2 + 4 \cdot 4 = 8 \Rightarrow x_2 = 8 \end{cases}$$

Итак, 4-й столбец:

$$\xi_4 = (5, 8, 2, 0, 4)^T$$

5-й столбец:

$$\begin{cases} x_1 = 10 \cdot x_2 + 5 \cdot x_4 + 3 \cdot x_5 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 7 \cdot x_3 + 4 \cdot x_5 \\ x_3 = 2 \cdot x_4 + 11 \cdot x_5 \\ x_4 = 6 \cdot x_3 + 3 \cdot x_5 \\ x_5 = 3 \cdot x_2 + 4 \cdot x_4 + 0 \Rightarrow x_5 = 0 \end{cases}$$

$$\begin{cases} x_1 = 10 \cdot x_2 + 5 \cdot x_4 + 3 \Rightarrow x_1 = 3 \\ x_2 = 3 \cdot x_1 + 5 \cdot x_2 + 7 \cdot x_3 + 4 \Rightarrow x_2 = 4 \\ x_3 = 2 \cdot x_4 + 11 \Rightarrow x_3 = 5 \\ x_4 = 6 \cdot x_3 + 3 \Rightarrow x_4 = 3 \end{cases}$$

$$\xi_5 = (3, 4, 5, 3, 0)^T$$

Имеет место уменьшение метки по пути из 3-й вершины в 5-ю через 4-ю: $2+3=5 < 11$.

Вся матрица кратчайших расстояний:

$$C = \begin{pmatrix} 0 & 6 & 11 & 5 & 3 \\ 3 & 0 & 7 & 8 & 4 \\ 11 & 8 & 0 & 2 & 5 \\ 9 & 6 & 6 & 0 & 3 \\ 6 & 3 & 10 & 4 & 0 \end{pmatrix} \quad A = \begin{pmatrix} +\infty & 10 & +\infty & 5 & 3 \\ 3 & 5 & 7 & +\infty & 4 \\ +\infty & +\infty & +\infty & 2 & 11 \\ +\infty & +\infty & 6 & +\infty & 3 \\ +\infty & 3 & +\infty & 4 & +\infty \end{pmatrix}$$

Исходный граф сильно связан.

См. также семинар №6 и презентацию «Пример задачи о путях».

3. Поиски

3.1. Дерево, лес.

Неориентированное дерево – это, по определению, любой связный ациклический неор. граф.

Легко показать, что в неориентированном дереве существует вершина, степень которой равна единице (доказать!). Такая вершина называется *листом*.

Неориентированный лес – любой ациклический граф. То есть это граф, в котором каждая компонента связности является деревом.

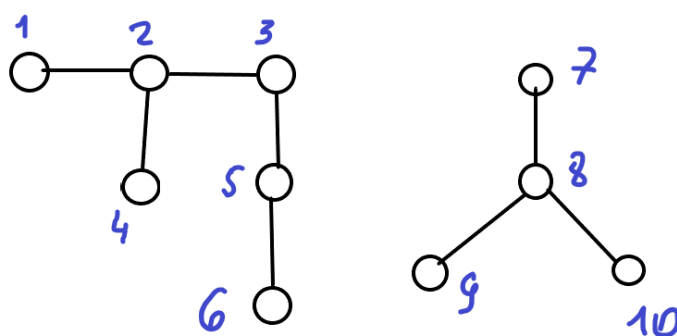


Рис. 1. Неориентированный лес

Ориентированное дерево (ордереве) – это бесконтурный орграф, в котором каждая вершина имеет полустепень захода, не большую единицы, и существует единственная вершина, называемая *корнем*, имеющая нулевую полустепень захода.

Легко вывести отсюда, что в ордереве каждая вершина достижима из корня (задача!).

Ориентированный лес – орграф, в котором каждая слабая компонента есть ордереве.

Рассмотрим некоторую терминологию по ориентированным деревьям, называемым дальше просто деревьями.

Вершина с нулевой полустепенью исхода называется *листом*.

Дерево есть лишь слабо связный граф (почему?), а компонента связности в нем – любой подграф, порожденный множеством вершин, лежащих на каком-либо пути, ведущем из корня в лист.

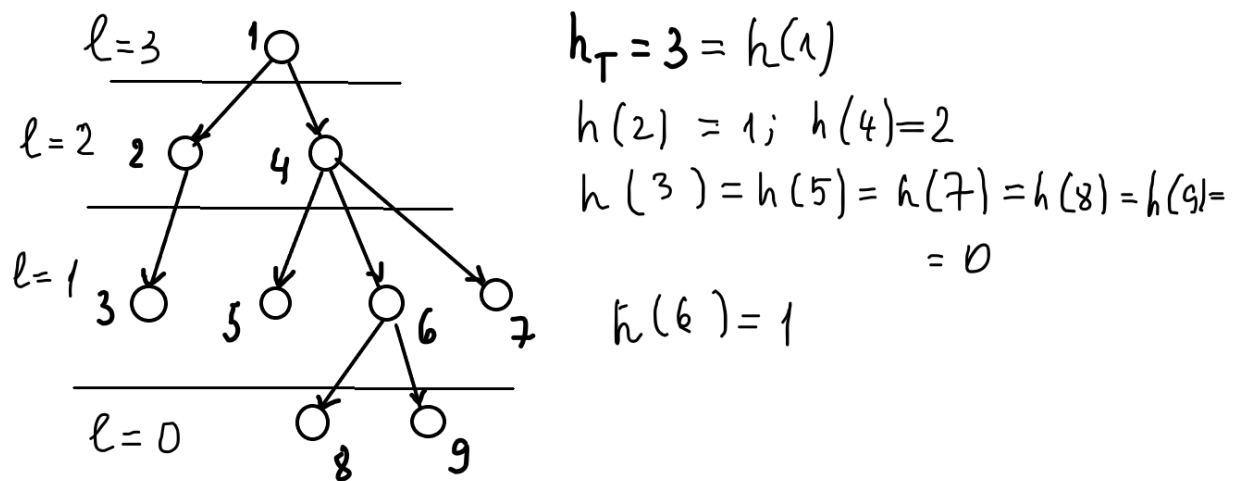


Рис. 2. Ориентированное дерево

Преемник любой вершины дерева называется её сыном, а сама она – отцом. То есть отец – предшественник сына.

Если в дереве $u \Rightarrow^* v$, то первая вершина называется предком второй, тогда как вторая – потомком первой. В случае пути ненулевой длины, то есть при $u \Rightarrow^+ v$, говорят о подлинном предке (подлинном потомке).

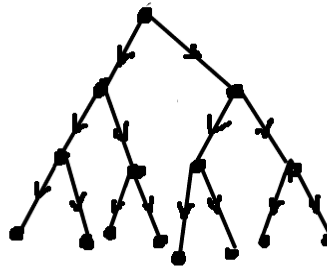
Каждая вершина u дерева характеризуется тремя числовыми параметрами: 1) $h(u)$ - высота вершины u - наибольшая длина пути из этой вершины в какой-то из листьев; при этом высота корня называется высотой всего дерева; 2) $d(u)$ - глубина вершины u - длина пути (единственного!) из корня в эту вершину; 3) $l(u)$ - уровень вершины u - разность между высотой дерева и глубиной данной вершины; то есть корень имеет наибольший уровень.

Дерево называется сбалансированным, если уровни всех его листьев одинаковы.

Дерево, сводящееся к корню и его сыновьям, называется кустом.

Введем понятие p -дерева. Так называется дерево, у которого полустепень исхода каждой вершины не больше $p \geq 2$. При $p = 2$ получаем бинарное дерево.

Полное p -дерево – это сбалансированное дерево, у которого полустепень каждой вершины, не являющейся листом, равна p .



$$h = 3$$

$$N = 2^3 = 8$$

Рис. 3. Полное бинарное дерево

По индукции легко показать, что число листьев в полном p -дереве равна $N = p^h$, где h - высота дерева. Тогда для любого p -дерева $N \leq p^h$. Логарифмируя, получим нижнюю оценку для высоты p -дерева с заданным числом листьев: $h \geq \log_p N$. Этот простой результат используется для нижней оценки сложности алгоритмов сортировки посредством сравнений. Тогда оценивается высота бинарного «дерева решений» с числом листьев $n!$ (где n – размерность сортируемого массива, а каждый лист дерева решений есть одна из перестановок его элементов):

$$N \geq \log_2 n! \approx \log_2 \left(\sqrt{2\pi n} \left(\frac{n}{e} \right)^n \right) = \log_2 (\sqrt{2\pi n}) + n(\log_2 n - \log_2 e) = O(n \log_2 n).$$

Использована известная формула Стирлинга для оценки факториала:

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e} \right)^n \text{ (при больших } n \text{)}.$$

Более подробно см. Учебник, п. 5.3.

3.2. Остов графа. Цикломатическое число неориентированного графа

Следующее понятие определяется для общего случая графа.

Остовом графа называется его максимальный подграф, являющийся лесом.

Ясно, что это должен быть остовный подграф, ибо, если это не так, то добавляем любую вершину исходного графа как изолированную и получаем подграф, являющийся лесом, что противоречит условию максимальности выбранного остова.

Может быть доказана такая теорема:

Теорема. Каждый граф имеет остов.

Доказательство для неор. графа см. в Учебнике, теорема 5.3.

Это чистая теорема существования, не дающая никаких способов построения остова. Далее мы опишем алгоритм поиска в глубину, который можно считать конструктивным доказательством сформулированной теоремы.

Теперь до конца этого параграфа рассматриваем только неор. графы и называем их просто графами.

Пусть у графа $G = (V, E)$ фиксирован какой-то остов $\mathfrak{T} = (V, T)$. Его ребра называют древесными, тогда как остальные – обратными (их множество обозначим буквой B).

Понятно, что добавление к остову любого обратного ребра приводит к появлению цикла. Среди всех циклов выделим такие, которые содержат только одно обратное ребро. Такие циклы называют *фундаментальными*. Таким образом, каждому фундаментальному циклу соответствует единственное обратное ребро. Но и наоборот, каждое обратное ребро может находиться только в одном фундаментальном цикле. Действительно, допуская, что какое-то обратное ребро находится сразу в двух фундаментальных циклах, то тут же получим, что существует цикл, проходящий только по древесным ребрам, что невозможно (рис. 4).

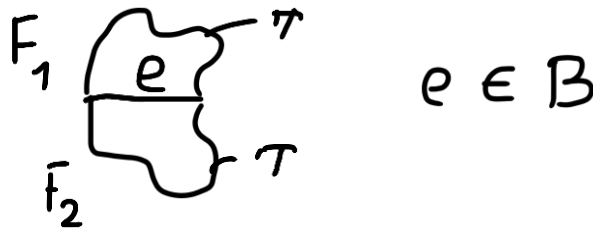


Рис. 4. Два фундаментальных цикла с общим обратным ребром (невозможная ситуация)

Итак, между множеством обратных ребер и множеством фундаментальных циклов установлено взаимно однозначное соответствие.

Обозначим $|E| = m, |T| = \mu, |B| = \nu$ число всех ребер, число древесных ребер и число обратных ребер (равное числу фундаментальных циклов) соответственно.

Число μ называется коциклическим рангом графа, число ν – циклическим рангом графа.

Для леса циклический ранг равен нулю, и он тем больше, чем граф богаче циклами.

Пусть далее, исходный граф имеет k компонент связности и пусть i -я компонента содержит n_i вершин ($i = 1, \dots, k$). Можно показать (см. семинар №5), что число ребер неор. дерева на единицу меньше числа его вершин.

Тогда общее число древесных ребер

$$|T| = \mu = (n_1 - 1) + (n_2 - 1) + \dots + (n_k - 1) = n - k.$$

Следовательно, циклический ранг равен

$$\nu = m - n + k.$$

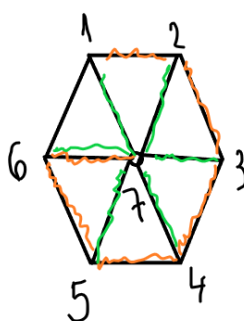
Эта простая формула очень важна. Она дает значение циклического ранга графа (так сказать, меры его цикличности) через параметры графа, никак не зависящие от того, как именно фиксирован остов. **Циклический ранг инвариантен относительно выбора остова.**

На следующем рисунке показаны два остова одного и того же графа. В каждом случае получим одно и то же:

$$v = 12 - 7 + 1 = 6.$$

Остов 1:

1-2, 2-3, 3-7, 4-5,
5-6, 6-7



$$V = 12 - 7 + 1 = 6$$

Остов 2:

1-7, 2-7, 3-7,
4-7, 5-7, 6-7

$$V = 6$$

$$n = 12, \mu = 6$$

Рис. 5. Два остова в неор. графе

На самом деле циклы графа образуют линейное пространство над полем \mathbb{Z}_2 , базисом которого является система фундаментальных циклов, а размерность равна циклическому рангу. Выбор остова равносильен выбору базиса. Мы вернемся к этому вопросу при рассмотрении алгоритма поиска в глубину¹.

3.3. Поиск в глубину в неориентированном графе

Списки смежности (Учебник, п. 5.2)

$\langle N[v], D[v], NEW[v], L[v] \mapsto \text{list}(u_1, \dots, u_l) \rangle$

Суть метода

Алгоритм ведет стек, в который помещаются вершины по мере обхода. Вершина покидает стек, как только оказывается пройденной некоторая ветвь поиска. Конкретный результат – разбиение множества ребер на древесные и обратные, а также перечисление фундаментальных циклов – зависит от выбора начальной вершины для поиска и порядка просмотра списка смежности каждой вершины. В конце работы стек должен быть пустым. Он может опустошаться не один раз, так как поиск ведется до тех пор, пока остаются неотмеченные вершины. Число опустошений стека дает число компонент связности в исходном графе.

¹ См. материал в Облаке под названием «Пространство циклов неориентированного графа».

Работу алгоритма поясним на примере:

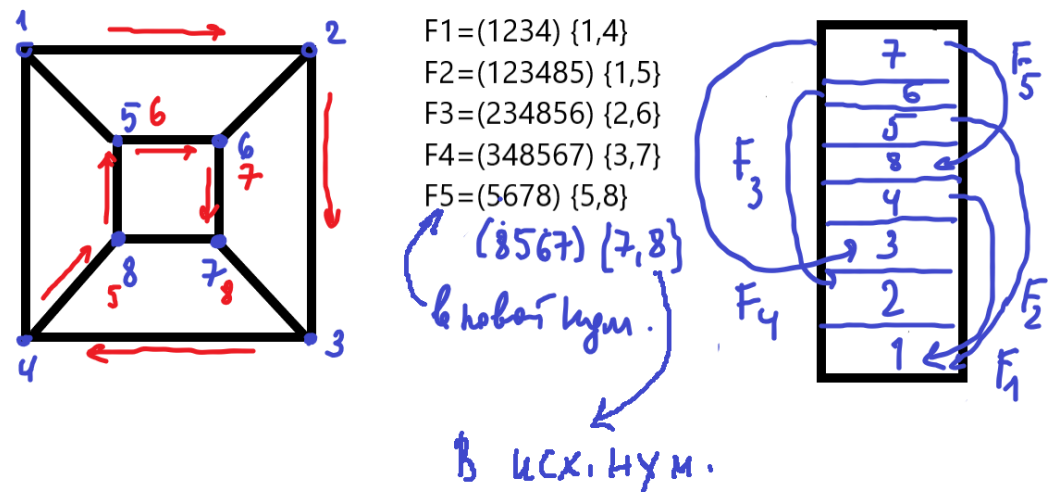


Рис. 6. Иллюстрация поиска в глубину в неориентированном графе

1 → (2, 4, 5)

2 → (1, 3, 6)

3 → (2, 4, 7)

4 → (1, 3, 8)

5 → (1, 6, 8)

6 → (2, 5, 7)

7 → (3, 6, 8_

8 → (4, 5, 7)

Начальная вершина – 1, списки смежности сканируем в порядке исходной нумерации вершин. В процессе поиска каждая вершина получает номер, который называют номером при поиске в глубину, или D-номером (depth search number). Этот номер может отличаться от исходного. На рисунке выше новые D-номера записаны красным цветом.

Процедура носит рекурсивный характер: так, стартуя и 1-й вершины, отмечаем ее, помещаем в стек, фиксируем ребро 1 – 2 как древесное и возобновляем поиск из 2-й вершины, откладывая возможные продолжения из 1-й вершины, и т. д. В данном примере дерево поиска не ветвится, но вот простой пример, когда возникает ветвление:

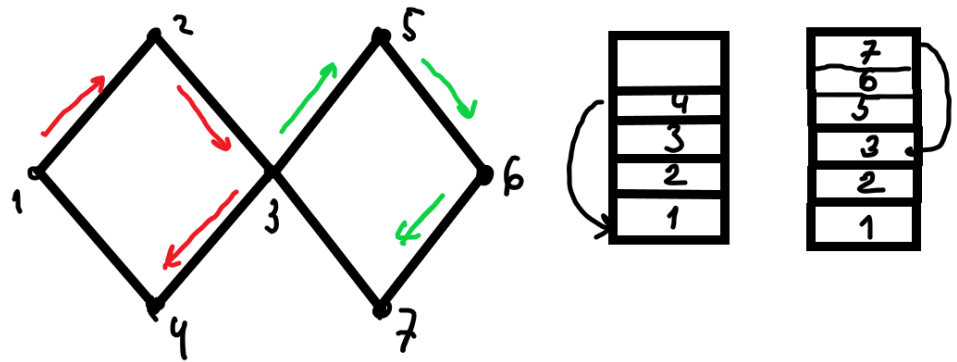


Рис. 7. Ветвящееся дерево поиска

Лекция №14

06.11.24

3.3. Поиск в глубину в неориентированном графе (продолжение)

Псевдокод

Вход: неориентированный граф $G = (V, E)$, заданный списками смежности.

Выход: разбиение множества ребер $E = T \cup B$; D-нумерация вершин; множество FC фундаментальных циклов.

begin

1. Начальные установки: $T, B, FC := \emptyset$; $STACK := \emptyset$; $Count := 1$; (счетчик вершин).
2. **for all** $v \in V$ **do** $NEW[v] := 1$; **end**
3. **for all** $v \in V$ **while** $(\exists v)(NEW[v] = 1)$ **do** $SEARCH_D(v)$; **end**

proc $SEARCH_D(v)$

$NEW[v] := 0$;

$D[v] := Count$; $Count := Count + 1$;

$Push(v)$; (затолкнуть в стек)

for all $w \in L[v]$ **do**

if $NEW[w]$ **then begin**

$\{v, w\} \rightarrow T$;

SEARCH_D(w);

end

else if $\{v, w\} \notin T$ then if $\{v, w\} \notin B$ then begin

$\{v, w\} \rightarrow B$;

(здесь подразумевается, что обратное ребро добавляется к множеству B лишь тогда, когда его там нет)

Read(v...w) \rightarrow FC;

(чтение из стека без изменения содержимого вершин, лежащих на фундаментальном цикле)

end

end

Pop(v); (вытолкнуть из стека)

end SEARCH_D(v)

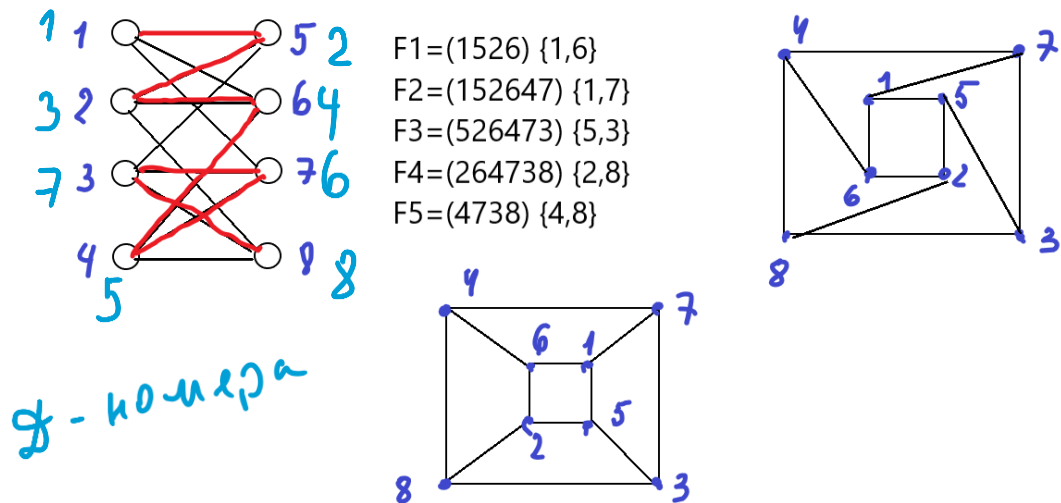


Рис. 6а. Поиск в глубину в неор. графе и преобразование графа.

Этот алгоритм действительно находит остов, так как: 1) каждая вершина инцидентна некоторому древесному ребру, 2) подграф (V, T) не имеет циклов по построению, 3) всякое ребро, не являющееся древесным, будет обратным и, следовательно, находится на некотором цикле, т.е. мы имеем максимальный подграф, являющийся лесом, т.е. остов. Компоненты связности этого леса будут *корневыми* неориентированными деревьями, в которых корнями будет очередная вершина, из которой возобновляется поиск.

Отметим также, что при поиске в глубину в неориентированном графе не могут появиться «поперечные» ребра, т.е. невозможно, чтобы некоторая вершина a , имела, по меньшей мере, двух сыновей (s_1 и s_2), между потомками которых есть ребро, не являющееся древесным.

Действительно, пусть вершины u и v – такие потомки.

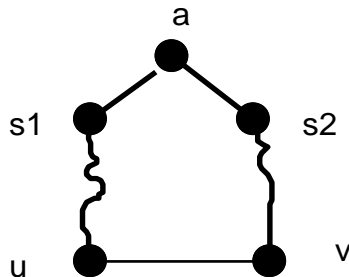


Рис.8. «Поперечное» ребро (которого не может быть)

Тогда ребро $\{u, v\}$, как лежащее на цикле, должно быть обратным, так как все остальные ребра этого цикла древесные, но оно не может быть обратным, так как соединяет вершины, не являющиеся ни предком, ни потомком друг друга. Полученное противоречие доказывает, что ситуация, изображенная на рис., не может иметь места, т.е. «поперечные» ребра невозможны.

Остов графа, который находит поиск в глубину (как в неориентированном, так в ориентированном графе), называется *остовным лесом поиска в глубину*, или *глубинным остовным лесом*.

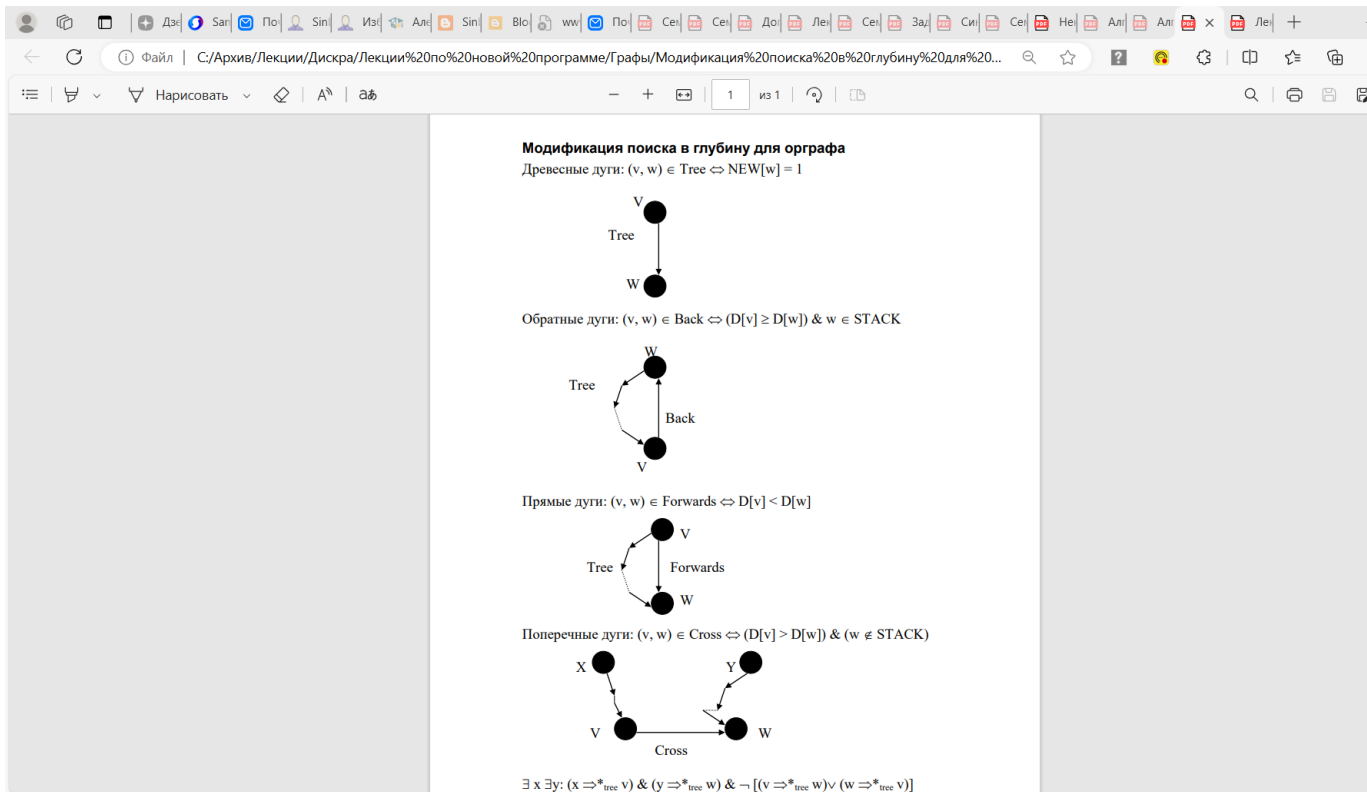
3.4. Поиск глубину в орграфе

Определение классов дуг

- 1) *Древесные дуги* (множество T): древесная дуга ведет от отца к сыну в глубинном остовном лесу.
- 2) *Обратные дуги* (множество B): обратная дуга ведет от потомка к предку в глубинном остовном лесу (в том числе все петли).
- 3) *Прямые дуги* (множество F): прямая дуга ведет от подлинного предка к подлинному потомку, но не от отца к сыну, в глубинном остовном лесу.
- 4) *Поперечные дуги* (множество C): дуга, не принадлежащая ни к одному из перечисленных выше множеств.

Более содержательная характеристика поперечной дуги состоит в том, что начало и конец такой дуги взаимно недостижимы в глубинном остовном лесу (то есть по древесным дугам). Они находятся либо на разных «ветвях» (компонентах связности) одного дерева в глубинном остовном лесу, либо на разных деревьях (слабых компонентах) этого леса.

(См. файл «Модификация поиска в глубину для орграфа» в Облаке.)



Соотношение D-номеров начала и конца дуги, принадлежащей тому или иному классу, показано ниже в псевдокоде алгоритма.

Первый пример:

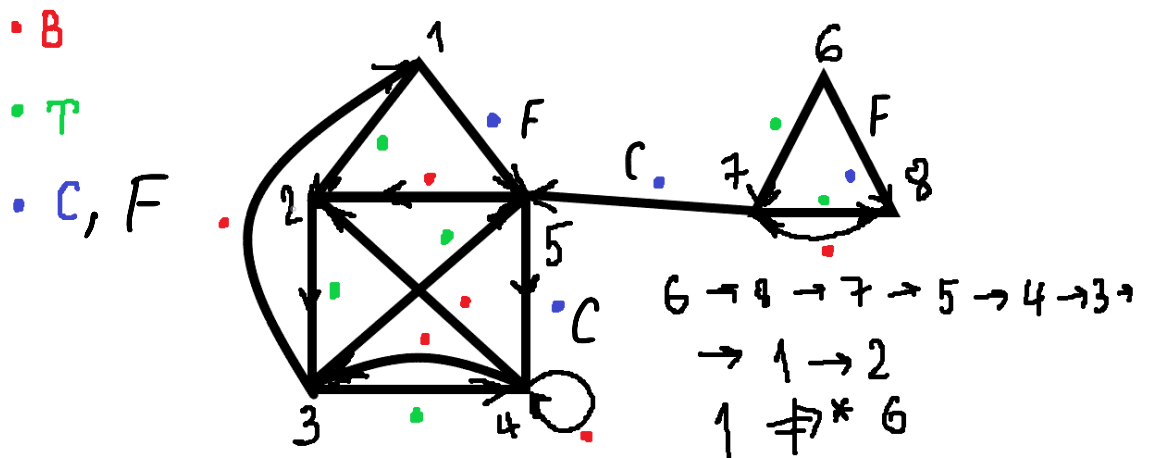


Рис. 9. Иллюстрация поиска в глубину в орграфе.

Псевдокод

Вход: ориентированный граф $G = (V, E)$, заданный списками смежности.

Выход: разбиение множества дуг $E = T \cup B \cup F \cup C$; D-нумерация вершин.

begin

4. Начальные установки: $T, B, F, C := \emptyset$; $STACK := \emptyset$; $Count := 1$; (счетчик вершин).
5. **for all** $v \in V$ **do** $NEW[v] := 1$; **end**
6. **for all** $v \in V$ **while** $(\exists v)(NEW[v]=1)$ **do** $SEARCH_DOR(v)$; **end**
end

proc $SEARCH_DOR(v)$

$NEW[v] := 0$;

$D[v] := Count$; $Count := Count + 1$;

$Push(v)$;

for all $w \in \Gamma(v)$ **do**

if $NEW[w]$ **then begin**

$(v, w) \rightarrow T$;

$SEARCH_DOR(w)$;

end

else begin

if $(D[v] \geq D[w]) \ \& \ (w \in STACK)$ **then** $(v, w) \rightarrow B$;

(обратные дуги)

if $(D[v] < D[w])$ **then** $(v, w) \rightarrow F$;

(прямые дуги)

if $(D[v] > D[w]) \ \& \ (w \notin STACK)$ **then** $(v, w) \rightarrow C$;

(поперечные дуги)

end

end

$Pop(v)$;

end $SEARCH_DOR(v)$

Критерий бесконтурности

Пример

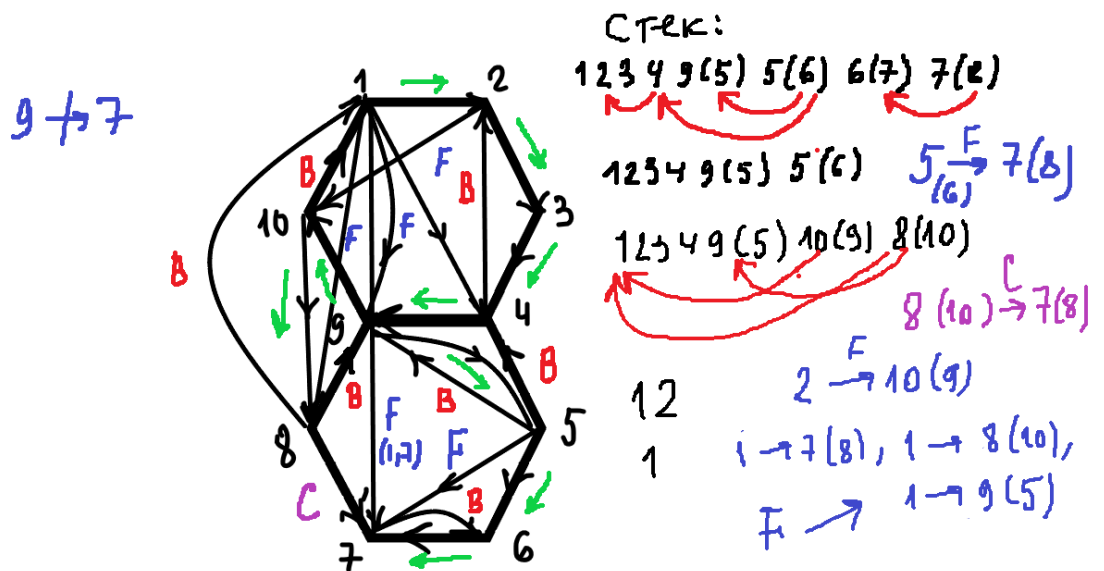


Рис. 10. Иллюстрация поиска в глубину в орграфе

Можно доказать такой критерий бесконтурности орграфа:

Орграф является бесконтурным тогда и только тогда, когда в результате поиска в глубину из произвольно фиксированной начальной вершины множество обратных дуг оказывается пустым.

3.5. Поиск в ширину

Алгоритм поиска в ширину в графе с числовыми метками дуг

Напомним сначала обычный алгоритм волнового фронта.

Алгоритм A0

Proc Search_B;

for all $v \in V$ **do** $M[v] := \infty$; **end**

$Q := \emptyset$;

$M[v_0] := 0$;

$v_0 \rightarrow Q$;

for all $v \in Q$ **while** $(Q \neq \emptyset)$ **do**

for all $w \in L[v]$ **do**

if $M[w] = \infty$ **then begin**

$w \rightarrow Q$;

$M[w] := M[v] + 1$;

end

end

$Q \rightarrow v$;

end

end Proc Search_B

Теперь рассмотрим модификацию алгоритма для графа с неотрицательными числовыми метками дуг.

Алгоритм A1

Search1_B;

/ Модификация алгоритма волнового фронта для произвольных неотрицательных числовых меток дуг. IN – признак присутствия вершины в очереди. В отличие от классического алгоритма волнового фронта вершина может многократно попадать в очередь. */*

for all $v \in V$ **do** $M[v] := \infty$; $IN[v] := 0$; **end**

$Q := \emptyset$;

$M[v_0] := 0$;

$v_0 \rightarrow Q$; $IN[v_0] := 1$;

for all $v \in Q$ **while** ($Q \neq \emptyset$) **do**

for all $w \in L[v]$ **do**

$\Delta := \min(M[w], M[v] + \varphi(\langle v, w \rangle))$;

if ($\Delta < M[w]$) **then begin**

$M[w] := \Delta$;

if $IN[w] = 0$ **then begin**

$w \rightarrow Q$;

$IN[w] := 1$;

end

end

end

$Q \rightarrow v$; $IN[v] := 0$;

end

end Proc Search1_B

Следует обратить внимание на то, что величина Δ вычисляется на выделенном шаге для того, чтобы зафиксировать нетривиальное снижение метки и в зависимости от этого ставить вершину в очередь снова или нет. Поэтому нельзя было перевычисление метки задать командой

$M[w] := \min(M[w], M[v] + \varphi(\langle v, w \rangle)).$

Обобщение для полукольца с тривиальной итерацией

Пусть $S=(S, +, \cdot, 0, 1)$ идемпотентное полукольцо с тривиальной итерацией и естественным порядком, обозначенным \leq .

Тогда алгоритм поиска в ширину переписывается следующим образом.

Алгоритм A2

Proc Search2_B

for all $v \in V$ **do** $M[v] := 0$; $IN[v] := 0$; **end**

$Q := \emptyset$;

$M[v_0] := 1$;

$v_0 \rightarrow Q$; $IN[v_0] := 1$;

for all $v \in Q$ **while** $(Q \neq \emptyset)$ **do**

for all $w \in L[v]$ **do**

$\Delta := \sup(M[w], M[v] \cdot \varphi(\langle v, w \rangle));$

if $(\Delta > M[w])$ **then begin**

$M[w] := \Delta$;

if $IN[w] = 0$ **then begin**

$w \rightarrow Q$;

$IN[w] := 1$;

end

end

end

$Q \rightarrow v$; $IN[v] := 0$;

end

end Proc Search2_B

Исходный алгоритм поиска в ширину будет частным случаем для полукольца \mathbb{R}^+ .

Пример.

Рассмотрим поиск в ширину в графе, размеченном над полукольцом \mathbb{R}^+ (рис. 1).

Матрица:

$$\begin{pmatrix} \infty & 2 & \infty & \infty & 1 & \infty \\ \infty & \infty & 5 & \infty & \infty & \infty \\ \infty & \infty & \infty & 3 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 1 \\ \infty & \infty & 1 & \infty & \infty & \infty \end{pmatrix}$$

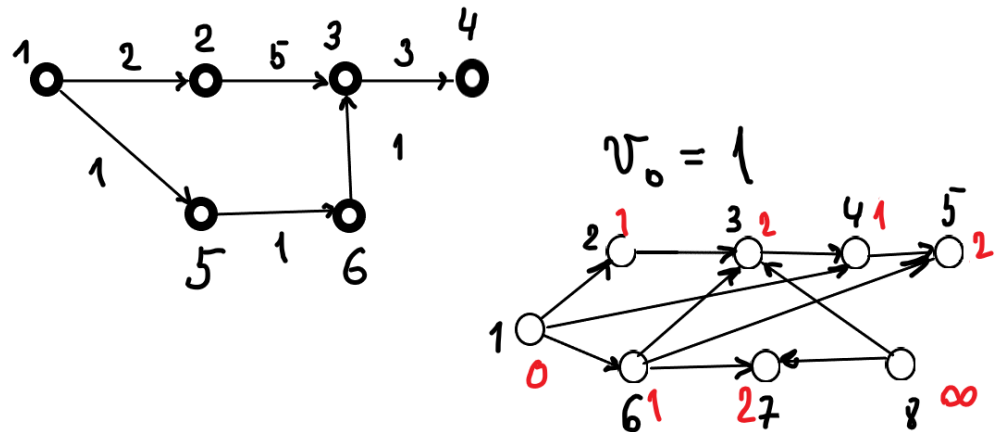


Рис. 11. Поиск в ширину

В данном случае формулы перевычисления меток имеют вид:

$$\Delta := \min(M[w], M[v] + \varphi(v, w));$$

$$\text{if } (\Delta < M[w]) \text{ then } (M[w] := \Delta)$$

Процесс работы алгоритма сведем в таблицу 1.

В этой таблице в колонке, описывающей изменения содержимого очереди, точкой с запятой мы отделяем текущую вершину поиска, пока еще находящуюся в очереди.

Таблица 1

1	Очередь Q	M[1]	M[2]	M[3]	M[4]	M[5]	M[6]
2	1	0	∞	∞	∞	∞	∞
3	1; 2, 5	0	2	∞	∞	1	∞
4	2; 5, 3	0	2	7	∞	1	∞
5	5; 3, 6	0	2	7	∞	1	2
6	3; 6, 4	0	2	7	10	1	2
7	6; 4, 3	0	2	3	10	1	2
8	4; 3	0	2	3	10	1	2
9	3	0	2	3	6	1	2
10	4	0	2	3	6	1	2
11	\emptyset	0	2	3	6	1	2

В строке 7 показано уменьшение метки вершины 3, и, поскольку она уже покинула очередь, ее необходимо туда вернуть. В строке 9 показано уменьшение метки вершины 4, и она тоже второй раз становится в очередь, но тут же покидает ее, так как множество ее преемников пусто.

Строка 1 показывает состояние до выполнения основного цикла.

4. Изоморфизм. Группа автоморфизмов

Определения и примеры

Графы $G_1 = (V_1, \rho_1)$ и $G_2 = (V_2, \rho_2)$ называются изоморфными, если существует биекция $h: V_1 \rightarrow V_2$ такая, что для любых двух вершин $u, v \in V_1$ $u \rho_1 v \Leftrightarrow h(u) \rho_2 h(v)$, то есть две вершины первого графа смежны тогда и только тогда, когда смежны их образы во втором.

Сама биекция h называется изоморфизмом графа $G_1 = (V_1, \rho_1)$ на граф $G_2 = (V_2, \rho_2)$.

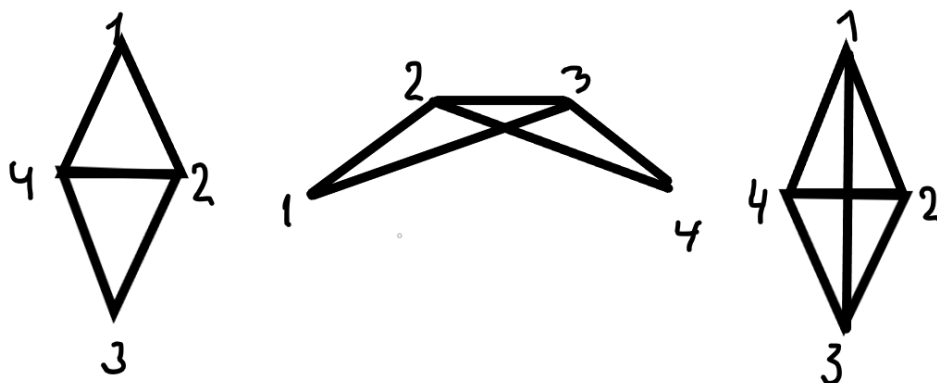


Рис. 12. Изоморфные и неизоморфные графы

На рис. 12 первые два графа изоморфны, а третий им не изоморфен, так как в нем 6 ребер, а в первых двух по 5.

Дальше в этом параграфе рассматриваются только неориентированные графы.

Проблема распознавания изоморфности графов в нашем курсе не рассматривается. Обсудим только некоторые примеры.

Сохранение степеней вершин

Необходимым (но не достаточным) условием изоморфности графов является сохранение при изоморфизме степеней всех вершин, то есть степень образа каждой вершины должна быть равна степени прообраза. Это легко вывести из определения.

Переход к дополнению

Иногда распознать то, что графы изоморфны, позволяет переход к дополнениям рассматриваемых графов.

Дополнением неориентированного графа $G=(V,E)$ называется граф $\bar{G}=(V,\bar{E})$, то есть две вершины в дополнении графа соединены ребром тогда и только тогда, когда они не соединены ребром в исходном графе. Понятно тогда, что два графа изоморфны тогда и только тогда, когда изоморфны их дополнения. Переходить к дополнению имеет смысл, если в нем существенно меньше ребер, чем в исходном графе.

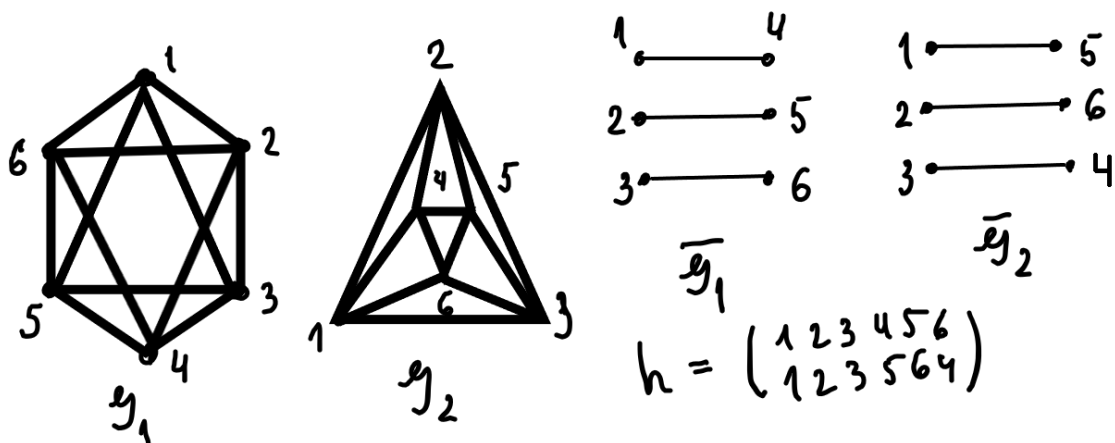


Рис. 13. Изоморфные графы

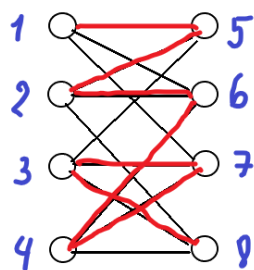
На рис. 13 исходные графы G_1 и G_2 4-регулярны, то есть степень каждой вершины равна 4. Это «почти полные» графы (в полном n -графе степень каждой вершины равна $n-1$), и поэтому в дополнениях степень каждой вершины будет равна $1=5-4$.

Изоморфность дополнений видна сразу.

Структура циклов. Использование поиска в глубину

В некоторых случаях установить изоморфность графов помогает поиск в глубину, обнаруживающий идентичность цикломатик (или отсутствие таковой).

На рисунках ниже D-номера не показаны. Сложение циклов понимается как выполнение операции симметрической разности на множествах их ребер.



$F1=(1526) \{1,6\}$
 $F2=(152647) \{1,7\}$
 $F3=(526473) \{5,3\}$
 $F4=(264738) \{2,8\}$
 $F5=(4738) \{4,8\}$

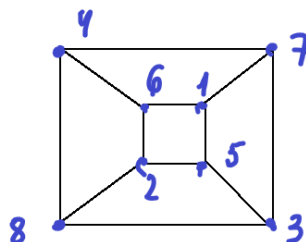
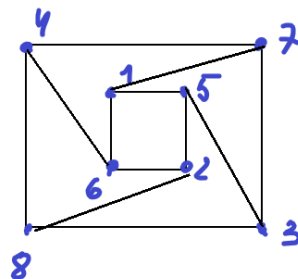
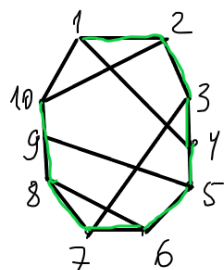
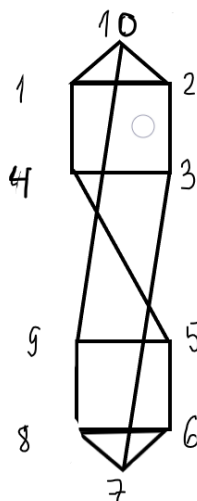


Рис. 14. Получение цикломатки графа с помощью поиска в глубину



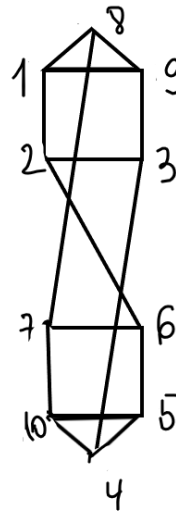
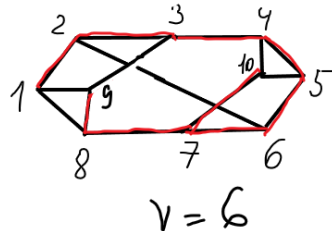
$$V = 6$$



Стек и фонд. циклы:
 $1234, 4 \rightarrow 1, F1=(1234)$
 $1234567, 7 \rightarrow 3, F2=(34567)$
 $12345678, 8 \rightarrow 6, F3=(678)$
 $123456789, 9 \rightarrow 5, F4=(56789)$
 $123456789 \ 10, 10 \rightarrow 1, 10 \rightarrow 2$
 $F5=(123456789 \ 10),$
 $F6=(23456789 \ 10)$
 $(12 \ 10) = F5 + F6$

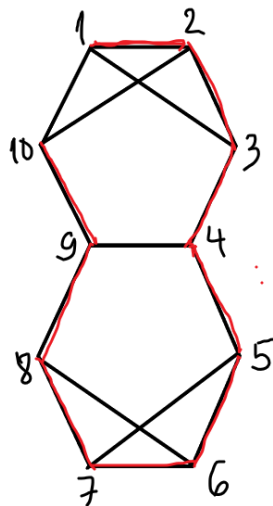
$$(5689) = F3 + F4$$

$F1=(23456)$, $L=5$
 $F2=(12345678)$, $L=8$
 $F3=(123456789)$, $L=9$
 $F4=(3456789)$, $L=7$
 $F5=(4567\ 10)$, $L=5$
 $F6=(567\ 10)$, $L=4$
 $F2+F3=(198)$
 $F5+F6=(45\ 10)$
 $F3+F4=(1239)$



Стек:
 123456, 2-6
 12345678, 1-8
 123456789, 1-9, 3-9
 1234567
 1234567 10, 4-10,
 5-10
 1234567
 123456
 12345
 1234
 123
 12
 1
 пусто

Рис. 15. Цикломатика и поиск в глубину



Стек
 123, 1-3
 1234567, 5-7
 12345678, 6-8
 123456789, 4-9
 123456789 10, 1-10, 2-10

$F5+F6=(12\ 10)$, $L=3$
 4 цикла длины 3

Фундаментальные циклы
 $F1=(123)$, $L=3$
 $F2=(567)$, $L=3$
 $F3=(678)$, $L=3$
 $F4=(456789)$, $L=6$
 $F5=(123456789\ 10)$, $L=10$
 $F6=(23456789\ 10)$, $L=9$

Рис. 16. Цикломатика и поиск в глубину

Первые два графа отказались изоморфными, так как их оба удалось преобразовать к одному и тому же графу, точнее к графам, совпадающим с точностью до номеров вершин (соответствующий изоморфизм легко записать:

$$h = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 9 & 3 & 2 & 6 & 5 & 4 & 10 & 7 & 8 \end{pmatrix}, \text{ и это не есть подстановка!}.$$

В каждом из этих графов есть два «домика»: цикл длины 3, «опирающийся» на цикл длины 4.

Третий граф им не изоморфен хотя бы потому, что в нем 4 цикла длины 3. Заметим, что все графы 3-регулярны.

Лекция №15

08.11.24

Аutomорфизмы графа

Аutomорфизм графа $G=(V,E)$ - это такая биекция $h:V \rightarrow V$ множества его вершин на себя, которая сохраняет ребра, то есть $(\forall u,v \in V)(u-v \Leftrightarrow h(u)-h(v))$. Другими словами, это такое преобразование множества вершин графа, которое сохраняет его как геометрическую фигуру.

Например, для квадрата



цикл $h=(1234)$ будет автоморфизмом (это не что иное, как поворот квадрата на 90 гр. по часовой стрелке), тогда как транспозиция $g=(12)$ автоморфизмом не будет, так как $g(2)=1-4=g(4)$, но ребра 2--4 нет.

Ясно, что композиция двух автоморфизмов есть автоморфизм, обратное к автоморфизму, также есть автоморфизм, и тождественное отображение, очевидно, автоморфизм. Следовательно, возникает группа, которая называется *группой автоморфизмов* данного графа. Эта группа для графа с n вершинами будет подгруппой всей симметрической группы степени n (группы S_n подстановок множества вершин графа). Поэтому, в силу теоремы Лагранжа, порядок группы автоморфизмов n -графа должен быть делителем числа $n!$. Например, у четырехвершинного графа не может быть 16 или 10 автоморфизмов, а может быть любое число, являющееся делителем числа 24.

Группу автоморфизмов графа G обозначают $Aut(G)$.

Проблема перечисления автоморфизмов произвольного графа может оказаться весьма сложной, и здесь мы рассмотрим несколько простых способов нахождения группы автоморфизмов неориентированных графов.

Прежде всего, для начала надо определить порядок группы автоморфизмов.

Он может быть найден из следующих соображений.

Пусть S - непустое конечное множество, а G - действующая на нем группа, т.е. некоторая подгруппа симметрической группы множества S . Элементы $s, s' \in S$ назовем **эквивалентными**, если для некоторой подстановки $\sigma \in G$ выполняется $s' = \sigma(s)$.

Легко проверяется, что тем самым действительно определено отношение эквивалентности на S . Обозначим его \sim_G .

Для произвольно фиксированного $s \in S$ рассмотрим множество G_s всех подстановок в G , для которых s является неподвижной точкой, т.е. $G_s = \{\sigma : \sigma(s) = s\}$.

Можно без труда показать, что G_s - подгруппа G ; она называется **стабилизатором** (или **стационарной подгруппой**) **элемента** s .

Обозначим через $G(s, s')$ множество всех подстановок, переводящих s в s' .

Могут быть доказаны следующие утверждения:

Утверждение 1. Число всех подстановок, которые переводят s в s' , равно порядку стабилизатора элемента s , т.е., $|G(s, s')| = |G_s|$.

Утверждение 2. Индекс $|G : G_s|$ подгруппы G_s в группе G равен числу элементов в классе эквивалентности $[s]_{\sim_G}$.

По теореме Лагранжа $|G| = |G : G_s| \cdot |G_s|$, и, согласно утверждению 2, $|G| = |[s]_{\sim_G}| \cdot |G_s|$.

Класс эквивалентности $[s]_{\sim_G}$ называют также **орбитой элемента** s .

Таким образом, **порядок группы автоморфизмов графа может быть вычислен как произведение порядка подгруппы-стабилизатора произвольно выбранной вершины на число вершин в ее орбите.**

Например, для нарисованного выше квадрата орбита вершины 1 состоит из всех вершин графа. Это следует хотя бы из того, что при поворотах на 90, 180 и 270 (или – 90) гр. по часовой стрелке вершина 1 «побывает» во всех остальных. Какие подстановки оставят вершину 1 на месте? Очевидно, кроме тождественной, только отражение относительно диагонали (1, 3), то есть транспозиция (24). Значит, квадрат имеет 8 автоморфизмов.

Найти эти автоморфизмы можно так.

Фиксируем подгруппу группы автоморфизмов, порожденную циклом (1234), то есть поворотом на 90 гр. по часовой стрелке. Это будет циклическая подгруппа

$$H = \{\varepsilon, (1234), (13)(24), (4321)\},$$

$$(24)H = (24)\{\varepsilon, (1234), (13)(24), (4321)\} = \{(24), (12)(34), (13), (14)(23)\}.$$

Группа автоморфизмов квадрата тем самым найдена полностью.

Замечание. Для квадрата это легко проверить, используя критерий нормальной подгруппы: подгруппа H группы G будет нормальной (нормальным делителем) тогда и только тогда, когда для любого элемента $a \in G$ имеет место равенство $aHa^{-1} = H$.

$$\begin{aligned}\sigma H \sigma^{-1} &= \sigma\{\varepsilon, (1234), (13)(24), (4321)\}\sigma^{-1} = \{\varepsilon, (1432), (13)(24), (1234)\}, \\ \sigma &\in \{(13), (24), (12)(34), (14)(23)\}.\end{aligned}$$

Рассмотрим более сложный пример (рис. 17).

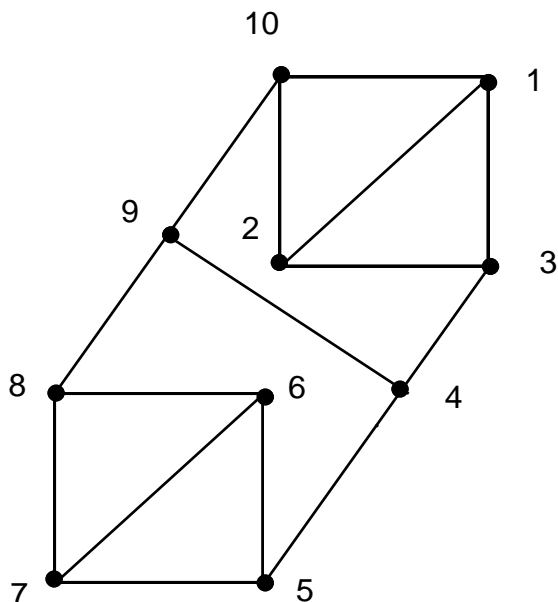


Рис. 17. Группа автоморфизмов графа

Возьмем вершину 1. Из соображений симметрии можно показать, что ее орбита состоит из нее самой, а также вершин 2, 6 и 7². Стабилизатор же ее, группа G_1 порождается подстановкой (67) и отражением (3 10)(49)(58), то есть

$$G_1 = [(67), (3\ 10)(49)(58)] = \{\varepsilon, (67), (3\ 10)(49)(58), (67)(3\ 10)(49)(58)\},$$

и порядок этой группы равен 4.

Таким образом, группа автоморфизмов нашего графа имеет порядок 16.

Можно тогда предложить такое ее разбиение на смежные классы.

В качестве «базовой» подгруппы удобно взять подгруппу, порождаемую транспозициями (12) и (67), то есть

$$H = \{\varepsilon, (12), (67), (12)(67)\}$$

Вычисляем еще три смежных класса по 4 элемента в каждом:

$$(17)(26)(35)(8\ 10)H,$$

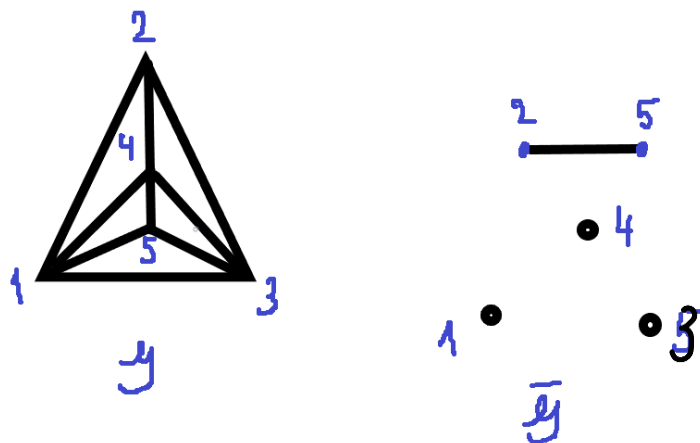
$$(3\ 10)(49)(58)H,$$

$$(17)(26)(35)(8\ 10)(3\ 10)(49)(58)H.$$

Рекомендуется закончить самостоятельно.

В некоторых случаях достаточно легко усматривается разложение группы в произведение своих подгрупп, не имеющих общих элементов, кроме единицы (тождественного автоморфизма).

Рассмотрим граф



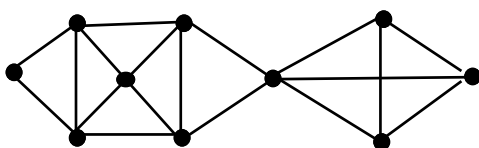
² Можно заметить, что вершины 1, 2 и 6, 7 являются (каждая пара) концами ребра, общего для двух циклов длины 3. Остальные вершины не обладают таким свойством. Поэтому эти вершины не могут быть переведены в остальные какими-то подстановками. Автоморфизм, как любой изоморфизм, должен сохранять всю цикломатику графа.

Рис. 18. Группа автоморфизмов графа и его дополнения

Рядом изображено его дополнение. Нетрудно показать, что группы автоморфизмов любого графа и его дополнения совпадают. В данном случае группа автоморфизмов дополнения получается как произведение своих подгрупп, каждая из которых действует независимо от другой на своем подмножестве. Именно, группа S_2 , действующая на подграфе $\{2, 5\}$, и группа S_3 , действующая на подграфе $\{1, 4, 3\}$ с пустым множеством ребер. Порядок всей группы равен 12, а каждый ее элемент есть произведение $\sigma\tau$, где $\sigma \in S_2, \tau \in S_3$, то есть

$$\text{Aut}(G) = \text{Aut}(\bar{G}) = \{\varepsilon, (25)\} \cdot \{\varepsilon, (134), (431), (13), (34), (14)\}.$$

Используя этот же подход, рекомендуется определить группу автоморфизмов следующего графа:



Как изменится решение, если в квадрате слева убрать центральную точку?

В некоторых случаях порядок группы автоморфизмов может быть вычислен из чисто комбинаторных соображений.

Вернемся к графам

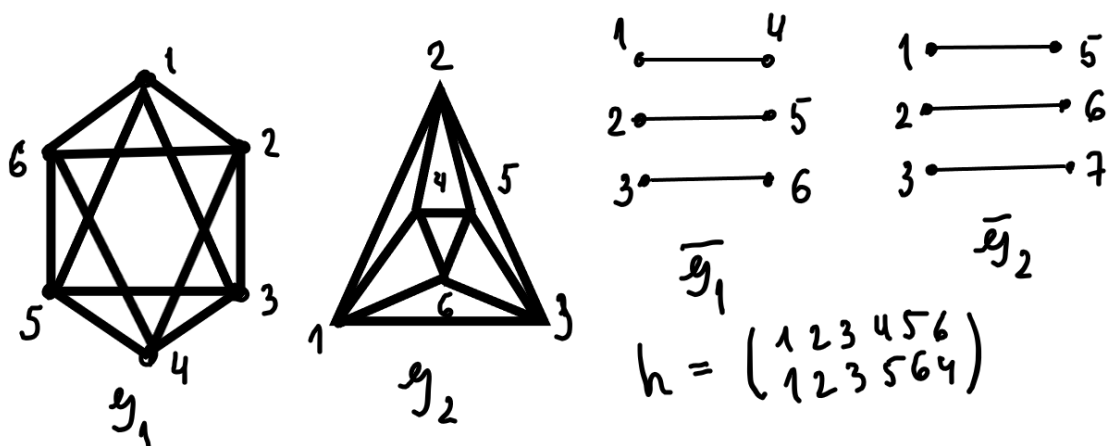


Рис. 19. Определение порядка группы автоморфизмов через переход к дополнению.

Это изоморфные графы, и порядок группы автоморфизмов легче найти для дополнений.

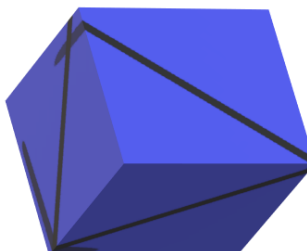
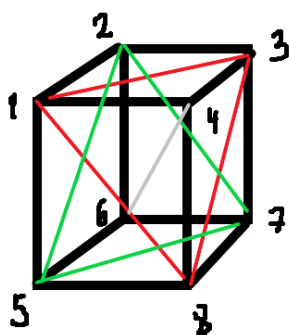
Три «спицы» (однореберных графа) можно переставлять 6 способами, не прибегая к внутренним отражениям (то есть транспозициям (14) , (25) и (36) – в дополнении первого графа). Затем можно выбрать какой-то подмножество «спиц» и применить в нем указанные отражения. Это можно сделать 8 способами (число подмножеств

трехэлементного множества). Следовательно, оба этих изоморфных графа имеют 48 автоморфизмов.

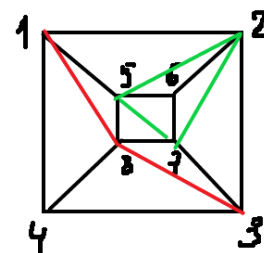
Это число можно получить и используя орбиты и стабилизаторы. Ясно, что каждая вершина тут может перейти во все остальные, то есть орбита любой вершины состоит из 6 элементов. Стабилизатором 1-й вершины будет группа автоморфизмов подграфа, порожденного множеством {2, 3, 5, 6}. Но это не что иное, как дополнение квадрата, построенного на этих вершинах, а порядок группы автоморфизмов квадрата равен 8. Следовательно, порядок группы автоморфизмов всего графа будет равен 48.

Вычисление всей группы тут весьма трудоемко, и мы его не приводим.

Дополнение. Некоторые автоморфизмы куба



Поворот вокруг
диагонали 46:
(275)(138)



Можно показать, что группа автоморфизмов куба (на множестве вершин) имеет порядок 48: 24 поворота и 24 отражения.

В теории многогранников доказывается, что группа поворотов правильного многогранника имеет порядок произведения числа граней на число вершин в грани.

https://ru.wikipedia.org/wiki/Правильный_многогранник