Содержание

1 Определение графа. Примеры графов. Степени вершин графа. Лемма о рукопожатиях

Определение. Граф (неориентированный) состоит из непустого конечно множества V и конечного множества E неупорядоченных пар элементов из V (записывается G = (V, E)).

Элементы множества $V = V_G$ называются **вершинами**, а элементы множества $E = E_G$ - **ребрами** графа G. Те и другие называются **элементами** графа.

Определение. Если $\{u, v\} \in E$, то будем записывать $e = uv$ и говорить, что вершины u и v смежны, а вершина u и ребро e инцидентны (так же, как вершина v и ребро e). Два ребра называются смежными, если они имеют общую вершину.

Определение. Степенью вершины v в графе G называется число ребер, инцидентных вершине v (обозначается $d_G(v) = d(v)$).

Вершина степени 0 - изолированная, вершина степени 1 - висячая. Минимальная и максимальная степени вершин графа G обозначаются $\delta(G), \Delta(G)$.

Последовательность степеней вершин графа G, выписанных в порядке неубывания называется степенной последовательностью или вектором степеней графа G.

Определение. Кратные ребра - два и более ребра, соединяющие одну и ту же пару вершин.

Определение. Петли - ребра, соединяющие вершины сами с собой.

Определение. Мультиграф - граф с кратными ребрами

Определение. Обыкновенный граф - граф без петель и кратных ребер.

Примеры графов:

1. Граф G = (V, E) с n вершинами и m ребрами называется (n, m)-графом, (1, 0)-граф называется тривиальным.
2. Пустой граф - O_n
3. Полный граф $K_n, C_n^2 = \frac{n(n-1)}{2}$
4. Двудольный граф $G = (V_1, V_2; E)$
5. Полный двудольный граф - $K_{p,q}$
6. Звезда - полный двудольный граф $K_{1,q}$
7. Простой цикл C_n
8. Регулярный (однородный) граф - граф, все вершины которого имеют одну и ту же степень. Кубические графы - 3-регулярные
9. Графы многогранников

Лемма 1.1 (О рукопожатиях). *Сумма степеней всех вершин произвольного графа $G = (V, E)$ - четное число, равное удвоенному числу его ребер:* $\sum_{v \in V} d_G(v) = 2|E|$

Доказательство. Индукция по числу ребер.
База: если в графе G нет ребер, то $\sum_{v \in V} d_G(v) = 0$.
Предположим, что формула верна для любого графа, число ребер в котором не превосходит $m \leq 0$.
Пусть $|E| = m + 1$. Рассмотри произвольное ребро $e = uv \in E$ и удалим его из графа G. Получим граф $G' = (V, E'), |E'| = m$. По предположению индукции $\sum_{v \in V} d_{G'}(v) = 2|E'| = 2m$
Тогда $\sum_{v \in V} d_G(v) = \sum_{v \in V} d_G(v) + 2 = 2m + 2 = 2|E|$. ▶

Теорема имеет место быть и для мультиграфов.

Следствие. В любом графе число вершин нечетной степени четно.

2 Маршруты, цепи, циклы. Лемма о выделении простой цепи. Лемма об объединении простых цепей

3 Эйлеровы графы. Критерий существования эйлерова цикла (теорема Эйлера)

4 Гамильтоновы графы. Достаточные условия существования гамильтонова цикла (теоремы Оре и Дирака)

5 Изоморфизм графов. Помеченные и непомеченные графы. Теорема о числе помеченных n-вершинных графов

6 Проблема изоморфизма. Инварианты графа. Примеры инвариантов. Пример полного инварианта

7 Связные и несвязные графы. Лемма об удалении ребра. Оценки числа ребер связного графа

8 Плоские и планарные графы. Графы Куратовского. Формула Эйлера для плоских графов

Графы Куратовского

Замечание. Графы $K_{3,3}$ и K_5 непланарны

Доказательство. $K_{3,2}$ - плоский, в нем по формуле Эйлера 3 грани независимо от способа изображения. Пытаемся добавить 6 вершину, подставляя ее в каждую грань, получаем каждый раз противоречие - невозможность соединить вершину с необходимыми. Аналогично для K_5. ▶

Теорема 8.1 (Формула Эйлера для плоских графов). *Для любого связного плоского графа G = (V, E) верно $n - m + l = 2$, где n = |V|, m = |E|, l - число граней*

Доказательство. Рассмотрим две операции перехода от связного плоского графа G к его связному плоскому подграфу, не изменяющие величины $n - m + l$

1. удаление ребра, принадлежащего сразу 2 граням (одна из которых может быть внешней) **уменьшает m и l на 1**
2. удаление висячей вершины (вместе с инцидентным ребром) **уменьшает m и n на 1**

Очевидно, что любой связный граф после этих операций может быть приведен к тривиальному, а для него формула верна $\implies$ верна и для данного ▶

9 Деревья. Теорема о деревьях (критерии)

Теорема 9.1 (о деревьях №1). *Для (n, m)-графа G следующие определения эквивалентны:*

1. *G - дерево*
2. *G - связный граф и $m = n - 1$*
3. *G - ациклический граф и $m = n - 1$*

Доказательство. 1 $\to$ 2 Дерево - связный, планарный граф (имеет 1 грань) $\implies$ $n - m + 1 = 2$ $\implies$ $m = n - 1$

2 $\to$ 3 Пусть граф не ациклический $\implies$ есть цикл и e - циклическое ребро. Тогда по лемме об удалении ребра граф $G - e$ также связен и имеет m - 1 = n - 2 ребер $\implies$ противоречие оценке числа ребер связного графа $\implies$ граф ациклический

3 $\to$ 1 Обозначим число компонент связности - k. Пусть T_i - iтая компонента, является (n_i, m_i)-графом. Т.к T_i - дерево, то по ранее доказанному (1 $\to$ 2) $m_i = n_i - 1, i = \overline{1, k}$ $\implies$ $n - 1 = m = \sum_{i=1}^{k} m_i = \sum_{i=1}^{k} n_i - k = n - k$ $\implies$ $k = 1$ $\implies$ граф связный ▶

Теорема 9.2 (о деревьях №2). *Для (n, m)-графа G следующие определения эквивалентны:*

1. *G - дерево*
2. *G - ациклический граф и если $\forall$ пару несмежных вершин соединить ребром, то полученный граф будет содержать ровно 1 цикл*
3. *$\forall$ 2 вершины графа G соединены единственной простой цепью*

Доказательство. 1 $\to$ 2 В связном графе все несмежные вершины соединены простой цепью **(по лемме о выделении простой цепи)** $\implies$ добавление ребра e = uv приведет к образованию цикла, а два цикла образоваться не может в силу свойства циклов

2 $\to$ 3 любые две несмежные вершины u,v графа G соединимы, иначе при добавлении ребра uv не появится цикл $\implies$ в силу леммы о выделении простой цепи любые две вершины соединены простой цепью. А она единственная, иначе по лемме об объединении простых цепей в графе G был бы цикл.

3 $\to$ 1 из условия следует, что граф связен, а существование цикла противоречит условию единственности цепи $\implies$ граф ациклический.

▶

Лемма 9.1 (О листьях дерева). *В любом нетривиальном дереве имеется не менее двух листьев*

Доказательство. $\forall v \in V \ d(v) \geq 1$

$\sum_{v \in V} = 2|E| = 2m = 2(n-1) = 2n - 2$

Если 2 листа - то у 2 вершин степень 1 и у остальных n-2 как минимум 2, а для меньшего количества листьев оценка суммы неверна

$\sum_{v \in V} \leq 2 + (n-2)2 = 2n - 2$ ▶

10 Перечисление деревьев. Теорема Кэли о числе помеченных n-вершинных деревьев

11 Центр дерева. Центральные и бицентральные деревья. Теорема Жордана

12 Изоморфизм деревьев. Процедура кортежирования. Теорема Эдмондса

Теорема 12.1 (Теорема Эдмондса). *Два дерева изоморфны $\Leftrightarrow$ совпадают их центральные кортежи*

Доказательство. $\Rightarrow$ Если деревья изоморфны, то любой изоморфизм биективно отображает V_1 на V_1' (листья каждого уровня, $V_2 = T - V_1$etc). Поэтому для соответствующих вершин совпадают уровень и их кортежи, в т.ч и центральные.

$\Leftarrow$ Пусть центральные вершины v и v' произвольных деревьев T и T' имеют одинаковые кортежи. По кортежу однозначно восстанавливаются кортежи смежных с v вершин низших уровней $(1, \ldots s)$, их количество и порядок следования, в поддереве T_v

это все вершины, смежные с v, то же самое справедливо для вершин из T'. Таким образом, из совпадения центральных кортежей следует совпадение кортежей для вершин $v_j, v'_j, j = 1, \dots s$. После применения к каждой паре невисячих вершин v_j, v'_jполучаем изоморфизм T_v, T'_v

Для центрального дерева доказательство закончено, а для бицентрального повторяем доказательство для второй пары центральных вершин.

►

13 Вершинная и реберная связность графа. Основное неравенство связности

14 Отделимость и соединимость. Теорема Менгера

15 Реберный вариант теоремы Менгера

16 Критерии вершинной и реберной k-связности графа (без доказательства)

17 Ориентированные графы. Основные понятия. Ормаршруты и полумаршруты. Ориентированые аналоги теоремы Менгера

Определение. Ориентированным (v_1, v_{k+1})-маршрутом (ормаршрутом) в орграфе G называется чередующаяся последовательность вершин и дуг

$$P = (v_1, e_1, \dots, e_k, v_{k+1})$$

в которой $e_i = v_i v_{i+1}$

Если в орграфе нет кратных дуг, то ормаршрут может быть задан последовательностью входящих в него вершин. В любом случае ормаршрут задается дугами.

Определение. Орцепь - ормаршрут, в котором все дуги различны.

Определение. Простая орцепь (путь) - ормаршрут, в котором все вершины различны.

Определение. Орцикл - замкнутая орцепь.

Определение. Контур - замкнутый путь.

Определение. Полумаршрут - последовательность вершин и дуг орграфа, если для $\forall i = \overline{1,k}$ $e_i = v_i v_{i+1} \in E$ $\vee e_i = v_{i+1} v_i \in E$

Аналогично определяются **полупуть и полуконтур**.

Определение. Орграф называется обыкновенным, если он не содержит петель и кратных дуг.

Определение. Орграф называется направленным, не имеющий симметричных пар дуг.

Определение. Основание орграфа G - неориентированный орграф, полученный снятием ориентации всех дуг.

18 Ориентированные графы. Достижимость и связность. Три типа связности. Критерии сильной, односторонней и слабой связности орграфа

Тривиальный орграф является сильным, односторонним и слабым.

19 **Основные структуры данных для представления графов в памяти компьютера. Их достоинства и недостатки**

20 **Влияние структур данных на трудоемкость алгоритмов (на примере алгоритма отыскания эйлерова цикла)**

21 **Задача о минимальном остовном дереве. Алгоритм Прима**

22 **Задача о кратчайших путях. Случай неотрицательных весов дуг. Алгоритм Дейкстры**

23 **Потоки в сетях. Увеличивающие пути. Лемма об увеличении потока**

24 **Алгоритм Эдмондса-Карпа построения максимального потока**

25 **Разрезы. Лемма о потоках и разрезах. Следствие**

26 **Теорема Форда-Фалкерсона**

27 **Два критерия максимальности потока.**

28 **Приложения теории потоков в сетях. Задачи анализа структурно-надежных коммуникационных сетей**

29 **Задачи комбинаторной оптимизации. Массовая и индивидуальная задачи. Трудоемкость алгоритма. Полиномиальные и экспоненциальные алгоритмы**

30 **Задачи распознавания свойств. Детерминированные и недетерминированные алгоритмы. Классы P и NP. Проблема “P**