Вопросы к экзамену по Теории Графов и Комбинаторным Алгоритмам
3 семестр

Данил Заблоцкий

19 марта 2024 г.

Содержание

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

Определение 1 (Неориентированный граф, вершины и ребра графа). *Неориентированный граф* – пара множеств $G = (V, E)$, где

V – непустое конечное множество,
E – множество, состоящее из неупорядоченных пар элементов из V.

Элементы множества V называются *вершинами*, а элементы E – *ребрами* графа.

Примечание. Если $u, v \in V$, $\{u, v\} \in E$, то будем записывать

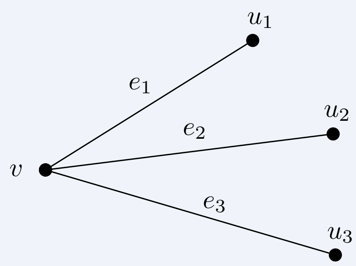
$$e = uv \ (= vu)$$

и говорить, что вершины u и v *смежны*, вершина u и ребро e – *инцидентны*.

Определение 2 (Степень вершины). *Степенью вершины* v называется число инцидентных ей ребер.

Обозначение: $d(v)\ \big(deg(v)\big)$

Пример. $deg(v) = 3$



Пример. *Пустой граф* – граф без ребер: O_n.

Пример. *Полный граф* – граф, любая пара которого смежна: K_n.

Примечание.

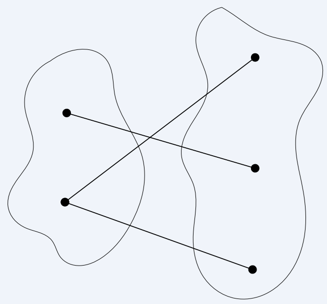
$$|E| = C_n^2 = \frac{n(n-1)}{2} \text{ – число ребер.}$$

Пример. *Двудольный граф* – граф, вершины которого разбиты на 2 непересекающиеся части (доли) так, что любое ребро ведет из одной доли в другую.

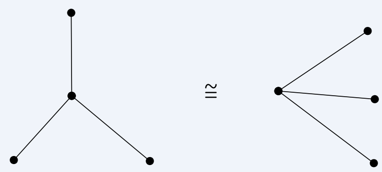
Если любая вершина одной доли смежна с любой вершиной другой доли, то такой граф называется *полным двудольным*.

Полный двудольный граф с долями размера p и q обозначают: $K_{p,q}$,

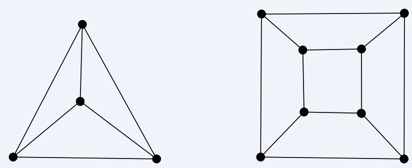
$$|E| = p \cdot q.$$



Пример. *Звезда* – полный двудольный граф $K_{1,q}$: одна доля состоит из одной вершины, а из нее веером расходятся лучи.



Пример. Графы многогранников



тетраэдр куб

Лемма 1 (О рукопожатиях)**.** Пусть $G = (V, E)$ – произвольный граф. Сумма степеней всех вершин графа G – четное число, равное удвоенному количеству его ребер:

$$\sum_{v \in V} deg_G(v) = 2|E| \tag{1}$$

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

Определение 3 (Маршрут)**.** *Маршрутом*, соединяющим вершины u и v ((u, v)-маршрут), называется чередующаяся последовательность вершин и ребер вида

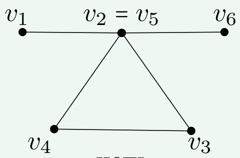
$$(u = v_1, e_1, v_2, \dots, v_k, e_k, v_{k+1} = v)$$

такая, что $e_i = v_i v_{i+1},\ i = \overline{1, k}$.

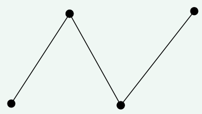
Определение 4 (Замкнутый маршрут)**.** Маршрут называется *замкнутым*, если первая вершина совпадает с последней, то есть

$$v_1 = v_{k+1}.$$

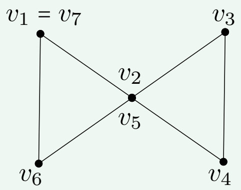
Определение 5 (Цепь, простая цепь)**.** Маршрут называется *цепью*, если в нем все ребра различны и *простой цепью*, если в нем все вершины различны (за исключением, быть может, первой и последней).



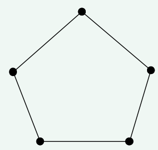
цепь


простая цепь

Определение 6 (Цикл, простой цикл)**.** Замкнутая цепь называется *циклом*, а замкнутая простая цепь – *простым циклом*.



цикл


простой цикл

Лемма 2 (О выделении простой цепи)**.** Всякий незамкнутый (u, v)-маршрут содержит простую (u, v)-цепь.

Лемма 3 (Об объединении простых цепей)**.** Объединение двух несовпадающих простых (u, v)-цепей содержит простой цикл.

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

Определение 7 (Эйлеров цикл)**.** Пусть $G = (V, E)$ – произвольный граф (мультиграф). Цикл в графе G называется *эйлеровым*, если он содержит все ребра графа.

Определение 8 (Эйлеров граф)**.** Граф называется *эйлеровым*, если в нем есть эйлеров цикл.

Теорема 1 (Эйлер, 1736)**.** В связном графе $G = (V, E)$ существует эйлеров цикл $\Leftrightarrow$ все вершины графа G четны (то есть имеют четную степень).

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

Определение 9 (Гамильтонов цикл, граф)**.** Пусть $G = (V, E)$ – обыкновенный граф, $|V| = n$. Простой цикл в графе G называется *гамильтоновым*, если он проходит по всем вершинам графа.

Граф называется *гамильтоновым*, если он содержит гамильтонов цикл.

Определение 10 (Гамильтонова цепь)**.** Простая цепь в графе G называется *гамильтоновой*, если она проходит по всем вершинам графа.

Теорема 2 (Оре, 1960)**.** Пусть $n \geqslant 3$. Если в n-вершинном графе G для любой пары несмежных вершин u, v выполнено условие

$$deg(u) + deg(v) \geqslant n,$$

то граф – гамильтонов.

Теорема 3 (Дирак, 1953)**.** Пусть $n \geqslant 3$. Если в n-вершинном графе G для любой вершины выполнено условие

$$deg(v) \geqslant \frac{n}{2},$$

то граф – гамильтонов.

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

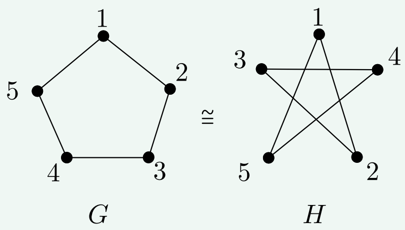
Определение 11 (Изоморфные графы). Графы $G = (V_G, E_G)$, $H = (V_H, E_H)$ называются *изоморфными*, если между множествами из вершин существует взаимнооднозначное соответствие

$$\phi : V_G \to V_H,$$

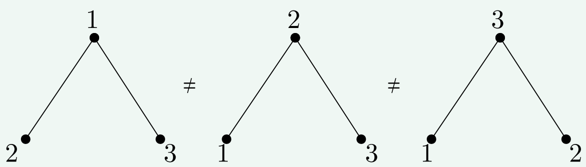
сохраняющее смежность, то есть $\forall u, v \in V_G$

$$uv \in E_G \Leftrightarrow \phi(u)\phi(v) \in E_H.$$

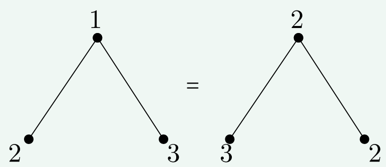
Обозначение: $G \cong H$



Определение 12 (Помеченный граф). Граф называется *помеченным*, если его вершины отличаются одна от другой какими-то метками.



3 разных помеченных графа



2 одинаковых помеченных графа

Теорема 4 (О числе помеченных n-вершинных графах)**.** Число p_n различных помеченных n-вершинных графов с фиксированным множеством вершин равно

$$2^{\frac{n(n-1)}{2}}.$$

6 Проблема изоморфизма. Инварианты графа. Примеры.

Определение 13 (Инвариант графа)**.** *Инвариант графа* $G = (V, E)$ – это число, набор чисел, функция или свойство связанные с графом и принимающие одно и то же значение на любом графе, изоморфном G, то есть

$$G \cong H \Rightarrow i(G) = i(G).$$

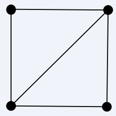
Инвариант i называется *полным*, если

$$i(G) = i(H) \Rightarrow G \cong H.$$

Обозначение: $i(G)$

Пример.

1. $n(G)$ – число вершин.
2. $m(G)$ – число ребер.
3. $\delta(G)$ – min степень.
4. $\Delta(G)$ – max степень.
5. $\phi(G)$ – плотность графа G – наибольшее число попарно смежных вершин.
6. $\varepsilon(G)$ – неплотность – наибольшее число попарно несмежных вершин.
7. $ds(G)$ – вектор степеней (или степенная последовательность) – последовательность степеней всех вершин, выписанная в порядке неубывания.
8. $\chi(G)$ – хроматическое число – наименьшее число χ, для которого граф имеет правильную χ-раскраску множества вершин (правильная раскраска – раскраска, при которой смежные вершины имеют разный цвет).

$Q_4 :$ 

$$\begin{array}{ll} n(Q_4) = 4 & \phi(Q_4) = 3 \\ m(Q_4) = 5 & \varepsilon(Q_4) = 2 \\ \delta(Q_4) = 2 & ds(Q_4) = (2,2,3,3) \\ \Delta(Q_4) = 3 & \chi(Q_4) = 3 \end{array}$$

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

Определение 14 (Соединимые вершины, связный граф)**.** Две вершины u, v графа G называются *соединимыми*, если в G $\exists$ (u, v)-маршрут.

Граф называется *связным*, если в нем любые две вершины соединимы.

Замечание. Тривиальный граф считается связным.

Определение 15 (Циклическое, ациклическое ребро)**.** Ребро e называется *циклическим*, если оно принадлежит некоторому циклу, и *ациклическим* – в противном случае.

Лемма 4 (Об удалении ребра)**.** Пусть $G = (V, E)$ – связный граф, $e \in E$.

1. Если e – циклическое ребро, то граф $G - e$ – связен.
2. Если e – ациклическое, то граф $G - e$ имеет ровно две компоненты связности.

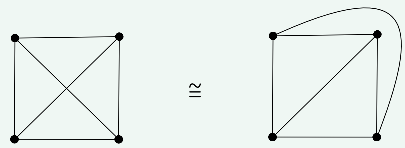
Теорема 5 (Оценки числа ребер связного графа)**.** Если G – связный (n, m)-граф, то

$$n - 1 \leqslant m \leqslant \frac{n(n-1)}{2}.$$

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

Определение 16 (Плоский, планарный граф). *Плоский граф* – это такой граф, вершины которого являются точками плоскости, а ребра – непрерывными плоскими линиями без самопересечений, соединяющими вершины так, что никакие два ребра не имеют общих точек вне вершин.

Планарный граф – это граф, изоморфный некоторому плоскому графу.

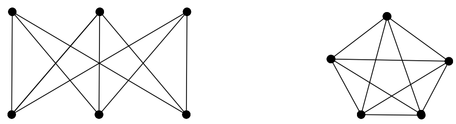
K_4 : 

$\cong$

не плоский, но планарный

плоский

Замечание. Несложно доказать, что графы $K_{3,3}$ и K_5 – непланарны.



$K_{3,3}$ K_5

Определение 17 (Гомеоморфные графы). Два графа называются *гомеоморфными*, если их можно получить из одного и того же графа с помощью разбиения ребер, то есть замены некоторых ребер простыми цепями.

Теорема 6 (Понтрягин-Куратовский). Граф планарен $\Leftrightarrow$ он не содержит подграфов, гомеоморфных $K_{3,3}$ или K_5.

Определение 18 (Грань). *Гранью* плоского графа называется максимальное множество точек плоскости, каждая пара из которых может быть соединена непрерывной плоскоской линией, не пересекающей ребер графа.

Теорема 7 (Формула Эйлера). Для всякого связного плоского графа верна формула

$$n - m + l = 2, \qquad (2)$$

где n – число вершин, m – число ребер, l – число граней графа.

9 Деревья. Первая теорема о деревьях.

Определение 19 (Ациклический граф, дерево)**.** Граф называется *ациклическим*, если в нем нет цикла. Связный ациклический граф называется *деревом*.

Теорема 8 (Первая теорема о деревьях)**.** Для (n, m)-графа G следующие утверждения эквивалентны:

1. G – дерево, то есть связный ациклический граф.
2. G – связен и $m = n - 1$.
3. G – ациклический и $m = n - 1$.

10 Деревья. Вторая теорема о деревьях.

Теорема 9 (Вторая теорема о деревьях)**.** Для (n, m)-графа G следующие утверждения эквивалентны:

1. G – дерево, то есть связный ациклический граф.
4. G – ациклический и если любую пару его несмежных вершин соединить ребром, то полученный граф будет содержать ровно один цикл.
5. Любые две вершины графа G соединены единственной простой цепью.

11 Теорема Кэли о числе помеченных n-вершинных деревьев (с леммой).

Лемма 5. При $n \geqslant 2$ существует взаимнооднозначное соответствие между множеством всех помеченных n-вершинных деревьев с метками $1, 2, \ldots, n$ и множеством всех слов длины $n - 2$ в алфавите $\{1, 2, \ldots, n\}$.

Теорема 10 (А. Кэли, 1889)**.** Число различных помеченных деревьев с n вершинами равно

$$t_n = n^{n-2}.$$

12 Центр дерева. Центральные и бицентральные деревья. Теорема Жордана.

Примечание. $d(u, v)$ – *длина* самой короткой простой (u, v)-цепи (длина – число ребер).

Определение 20 (Эксцентриситет). *Эксцентриситет* вершины v – расстояние до самой удаленной от v вершины графа:

$$\varepsilon(v) = \max_{u \in V} d(v, u).$$

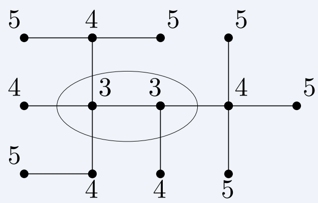
Определение 21 (Радиус). *Радиус* связного графа – это наименьший из эксцентриситетов его вершин:

$$\tau(G) = \min_{v \in V} \varepsilon(v).$$

Определение 22 (Центральная вершина). Вершина называется *центральной*, если ее эксцентриситет равен радиусу графа.

Определение 23 (Центр графа). Множество центральных вершин графа называется его *центром*.

Пример. Центр графа:



Определение 24 (Центральное, бицентральное дерево). Дерево, центр которого состоит из одной вершины, называется *центральным*, а дерево, центр которого состоит из двух смежных вершин – *бицентральным*.

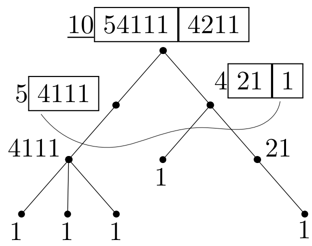
Теорема 11 (Жордан). Центр любого дерева состоит из одной или двух смежных вершин.

13 Изоморфизм деревьев. Процедура кортежирования (на примере). Теорема Эдмондса.

Примечание (Процедура кортежирования дерева).

Вход: n-вершинное дерево $T = (V, E)$.

Выход: Список натуральных чисел, представляющий кортеж T.



$\underline{3}11 > \underline{2}1$

$4\underline{2}11 > 4\underline{1}11$

Теорема 12 (Эдмондс)**.** Для изоморфизма деревьев необходимо и достаточно, чтобы совпадали их центральные кортежи.

14 Вершинная и реберная связность графа. Основное неравенство связности.

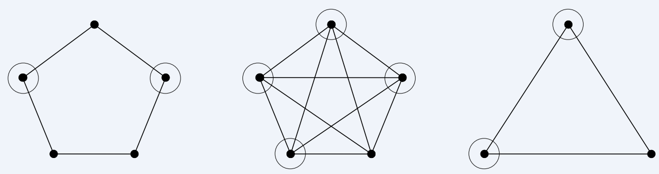
Определение 25 (Вершинная связность (связность))**.** *Вершинной связностью (связностью)* обыкновенного нетривиального графа G называется наименьшее число вершин, в результате удаления которых получается несвязный или тривиальный граф:

$$\mathcal{X}(G).$$

Примечание. Для тривиального графа по определению полагаем

$$\mathcal{X}(O_1) = 0.$$

Пример. Для C_5, K_5 и C_3

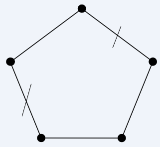


$\mathcal{X}(C_5) = 2$ $\mathcal{X}(K_5) = 4$ $\mathcal{X}(C_3) = 2$

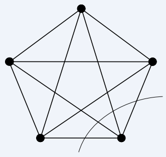
Определение 26 (Реберная связность)**.** *Реберной связностью* нетривиального графа называется наименьшее число ребер, в результате удаления которых получается несвязный граф:

$$\lambda(G).$$

Пример. $\lambda(O_1) = 0$,



$\lambda(C_5) = 2$



$\lambda(K_5) = 4$

Теорема 13 (Основное неравенство связности)**.** Для любого графа G

$$\varkappa(G) \leqslant \lambda(G).$$

15 Отделимость и соединимость. Теорема Менгера.

Определение 27 (Разделение вершин)**.** Пусть $G = (V, E)$ – связный граф, s и t – две несмежные вершины. Говорят, что множество вершин $\Omega \subset V$ *разделяет* s и t, если эти вершины принадлежат разным компонентам связности графа $G - \Omega$.

Определение 28 (k-отделимые вершины)**.** Несмежные вершины s и t называются *k-отделимыми*, если k равно наименьшему числу вершин, разделяющих s и t.

Определение 29 (Вершинно-независимые цепи)**.** Две простые цепи, соединяющие s и t, называются *вершинно-независимыми*, если они не имеют общих вершин, отличных от s и t.

Определение 30 (l-соединимые вершины)**.** Вершины s и t называются *l-соединимыми*, если l равно наибольшему числу вершинно-независимых цепей.

Теорема 14 (Менгер)**.** В связном графе любые две несмежные вершины k-отделимы $\Leftrightarrow$ они k-соединимы.

16 Реберный вариант теоремы Менгера.

Определение 31 (Разделение вершин)**.** Пусть $G = (V, E)$ – связный граф, s и t – две его произвольные вершины. Говорят, что множество ребер $R \subset E$ *разделяет* s и t, если эти вершины принадлежат разным компонентам связности графа $G - R$.

Определение 32 (k-реберно-отделимые вершины)**.** Вершины s и t называются *k-реберно-отделимыми*, если k равно наименьшему числу ребер, разделяющих s и t.

Определение 33 (Вершинно-независимые цепи)**.** Две простые цепи, соединяющие s и t, называются *вершинно-независимыми*, если они не имеют общих вершин, отличных от s и t.

Определение 34 (Реберно-независимые цепи)**.** Две простые цепи, соединяющие s и t, называются *реберно-независимыми*, если они не имеют общих ребер.

Определение 35 (l-реберно-соединимые вершины)**.** Вершины s и t называются *l-реберно-соединимыми*, если наибольшее число реберно-независимых (s,t)-цепей равно l.

Теорема 15 (Реберный аналог теоремы Менгера)**.** В связном графе любые две вершины k-реберно-отделимы $\Leftrightarrow$ они k-реберно-соединимы.

17 Критерии вершинной и реберной k-связности графа.

Следствие (Критерий вершинной k-связности графа)**.** Граф G k-связен $\Leftrightarrow$ любая пара его вершин соединена не менее, чем k вершинно-независимыми цепями.

Следствие (Критерий реберной k-связности графа)**.** Граф k-реберно-связен $\Leftrightarrow$ любая пара его вершин соединена не менее, чем k реберно-независимыми цепями.

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

Определение 36 (Ориентированный граф (орграф), вершины, дуги)**.** *Ориентированный граф (орграф)* G состоит из непустого конечного множества V и конечного множества $E \subset V \times V$ – упорядоченных пар элементов множества V:

$$G = (V, E).$$

Элементы множества V называются *вершинами*, а элементы множества E – *дугами* орграфа G.

Определение 37 (Ориентированный маршрут (ормаршрут), его длина, замкнутый ормаршрут)**.** Пусть $G = (V, E)$ – орграф. *Ориентированным маршрутом (ормаршрутом)* в орграфе G называется чередующаяся последовательность его вершин и дуг:

$$P = (v_1, e_1, v_2, e_2, \ldots, v_k, e_k, v_{k+1}),$$

в которой $e_i = v_i v_{i+1}$ – дуга орграфа G, $i = \overline{1, k}$.

Ормаршрут P также называется *ориентированным* (v_1, v_{k+1})-*маршрутом*.

Длина такого маршрута равна числу k его дуг.

Ормаршрут P называется *замкнутым*, если $v_1 = v_{k+1}$.

Определение 38 (Полумаршрут)**.** Последовательность

$$P = (v_1, e_1, v_2, e_2, \ldots, v_k, e_k, v_{k+1})$$

вершин и дуг орграфа $G = (V, E)$ называется *полумаршрутом*, если для любого $i = \overline{1, k}$ либо $e_i = v_i v_{i+1} \in E$, либо $e_i = v_{i+1} v_i \in E$.

Определение 39 (Слабо связный (слабый) орграф)**.** Орграф называется *слабо связным (слабым)*, если любые две его вершины соединены полумаршрутом.

Примечание (Доп. определения)**.** Для ориентированной теоремы Менгера:

Определение 40 (Вершинно-независимые (s, t)-пути)**.** Два (s, t)-пути называются *вершинно-независимыми*, если у них нет общих вершин, отличных от s и t.

Определение 41 ((s, t)-разделяющее множество вершин)**.** Множество W вершин орграфа G называется (s, t)-*разделяющим*, если в орграфе $G - W$ вершина t не достижима из s.

Теорема 16 (Ориентированная теорема Менгера)**.** Пусть $G = (V, E)$ – слабо связный орграф. Для любой пары вершин $s, t \in V$ таких, что $st \notin E$, наименьшее число вершин в (s, t)-разделяющем множестве равно наибольшему числу вершинно-независимых (s, t)-путей.

Примечание (Доп. определения)**.** Для реберного аналога ориентированной теоремы Менгера:

Определение 42 (Независимые по дугам (s,t)-пути)**.** Два (s,t)-пути называются *независимыми по дугам*, если они не имеют общих дуг.

Определение 43 ((s,t)-разделяющее множество дуг)**.** Множество R дуг орграфа G называется (s,t)*-разделяющим*, если в орграфе $G - R$ вершина t не достижима из s.

Теорема 17 (Реберный аналог ориентированной теоремы Менгера)**.** Пусть $G = (V, E)$ – слабо связный орграф. Для любой пары вершин $s, t \in V$ наименьшее число дуг в (s,t)-разделяющем множестве равно наибольшему числу независимых по дугам (s,t)-путей.

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

Примечание. Существует три различных понятия связности орграфа.

Определение 44 (Достижимая вершина)**.** Если в орграфе G существует ориентированный (u,v)-маршрут, то говорят, что вершина v достижима из вершины u.

Примечание. Любая вершина считается достижимой из самой себя.

Определение 45 (Сильно связный (сильный) орграф)**.** Орграф называется *сильно связным (сильным)*, если любые его две вершины взаимно достижимы.

Определение 46 (Односторонне связный (односторонний) орграф)**.** Орграф называется *односторонне связным (односторонним)*, если для любой пары его вершин хотя бы одна достижма из другой.

Определение 47 (Слабо связный (слабый) орграф)**.** Орграф называется *слабо связным (слабым)*, если любые две его вершины соединены полумаршрутом.

Определение 48 (Несвязный орграф)**.** Орграф называется *несвязным*, если он даже не является слабым.

Теорема 18 (Критерий сильной связности)**.** Орграф является сильно связным, если и только если в нем есть остовный замкнутый ормаршрут.

Теорема 19 (Критерий односторонней связности)**.** Орграф является односторонне связным, если и только если в нем есть остовный ормаршрут.

Теорема 20 (Критерий слабой связности)**.** Орграф является слабо связным, если и только если в нем есть остовный полумаршрут.

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

Примечание.

① **Матрица инцидентности** Это матрица с n строками, соответствующими вершинам, и m столбцами, соответствующими ребрам или дугам. Для неориентированного графа столбец, соответствующий ребру uv, содержит единицы в строках, соответствующих вершинам u, v, и нули в остальных строках. Для орграфа столбец, соответствующий дуге uv, содержит -1 в строке u, 1 в строке v и нули во всех остальных строках. Петлю, то есть дугу вида vv удобно представлять значением 2 в строке v.

Достоинства. Классический способ представления графа в теории.

Недостатки. С алгоритмической точки зрения, эта структура является самым худшим способом представления графа.

Во-первых, она требует порядка nm (то есть $\theta(nm)$) ячее памяти, большинство из которых занято нулями.

Во-вторых, неудобен доступ к информации. Ответ на элементарный вопрос типа «смежны ли некоторые вершины u, v?» или «существует ли вершина, смежная с данной вершиной v?» требует в худшем случае просмотра всех строки, то есть $O(m)$ шагов.

② **Матрица смежности** Это квадратная матрица $A = (a_{ij})$ размера $n \times n$, где

$$a_{ij} = \begin{cases} 1, & \text{если } v_i v_j \in E \\ 0 & \text{в противном случае.} \end{cases}$$

В неориентированном графе $v_i v_j \in E \Leftrightarrow v_j v_i \in E$, так что матрица смежности неориентированного графа симметрична, а для ориентированного графа - необязательно.

Достоинства. Прямой доступ к информации – возможность за один шаг получить ответ на вопрос «смежны ли некоторые вершины u, v?», а также удалить или добавить ребро uv.

Недостатки. Во-первых, ответ на вопрос «существует ли вершина, смежная с данной вершиной v?» требует в худшем случае просмотра всей строки, т.е. $O(n)$ операций.

Во-вторых, независимо от числа ребер и дуг графа объем занятой памяти составляет $\theta(n^2)$.

В-третьих, начальное заполнение матрицы смежности путем «естественной» процедуры имеет трудоемкость $\theta(n^2)$, что сразу сводит на нет алгоритмы линейной трудоемкости $O(n)$ при работе с графами, содержащими $O(n)$ ребер.

3 **Массив ребер и дуг**

Достоинства. Эта структура данных более предпочтительна по сравнению с 1 и 2 в смысле экономии памяти, если m $m \ll n^2$. Для хранения всего графа потребуется всего порядка m ячеек памяти.

Недостатки. Ответ на каждый из основых вопросов: «смежны ли некоторые вершины u, v?» или «существует ли вершина, смежная с данной вершиной v?» – требует в худшем случае просмотра всего массива, т.е. $O(m)$ шагов.

4 **Списки соседних вершин** Это динамическая структура данных, основанная на аппарате ссылочных переменных.

Для неориентированного графа она содержит для каждой вершины $v \in V$ список вершин, смежных с v. Каждый элемент списка является записью, содержащей информационное поле с меткой вершины u, смежной с v, и поле с указателем на следующий элемент списка.

Начало каждого списка хранится в массиве A ссылочных переменных, каждый элемент $A[v]$ которого является указателем на начало списка, содержащего вершины, смежные с вершиной v. Весь такой список вместе с указателем будем обозначать $A[v]$. Заметим, что для неориентированного графа каждое ребро uv представлено в списках дважды: элементом v в списке $A[u]$ и элементом u в списке $A[v]$.

Достоинства. Во-первых, для отыскания вершины, смежной с данной вершиной v, не нужно просматривать строку, как в матрице смежности, а достаточно лишь перейти по ссылке $A[v]$.

Во-вторых, число ячеек памяти, необходимое для представления графа посредством списков соседних вершин, имеет порядок $\theta(n+m)$.

В-третьих, это динамическая структура: при удалении ребра uv список $A[u]$ автоматически «сжимается», чем достигается экономия памяти.

Недостатки. Для удаления ребра uv требуется $O(n)$ операций: удалив элемент v списка $A[u]$, необходимо отыскать элемент u в списке $A[v]$, затратив в худшем случае n переходов по ссылке. Поэтому предпочтительнее использовать следующую модифицированную структуру данных.

5. **Списки соседних вершин с перекрестными ссылками** В этой структуре элемент v списка $A[u]$ содержит ссылку на элемент u списка $A[v]$, и наоборот.

Заметим, что для неориентированного графа каждое ребро uv представлено в списках дважды: элементом v в списке $A[u]$ и элементом u в списке $A[v]$.

Достоинства. Удаление ребра uv может быть выполнено за $O(1)$ операций (т.е. за число операций, ограниченное константой независимо от n). Для этого, удалив элемент v из списка $A[u]$, мы просто переходим по ссылке на элемент u списка $A[v]$ и удаляем его.

Недостатки. По всей видимости, лишен существенных недостатков по мнению В. П. Ильева, но проверять этот факт лень. Ну он профессор, так что, думаю, ему можно доверять на слово (хоть он и не говорил ни слова из этого раздела, лол).

6. **Списки соседних вершин для орграфов** В этой структуре $A[v]$ является указателем на начало списка, содержашего вершины, в которые ведут дуги из v.

Достоинства. Для орграфа каждая дуга uv представлена лишь один раз – элементом v в списке $A[u]$. Соответственно, удаление каждой дуги требует $O(1)$ операций.

Недостатки. При решении комбинаторных задач часто бывает нужно знать, какие дуги также и входят в вершину. Для этого приходится дополнительно использовать списки $B[v]$, содержащие вершины, из которых идут дуги в вершину v.

В ряде случаев вместо пары списков A, B для представления ориентированного графа предпочтительнее использовать *двумерный список*, в котором каждый элемент соответствует дуге uv и является как бы элементов сразу двух списков – «горизонтального» $A[u]$ и «вертикального» $B[v]$.

21 Задача о минимальном остовном дереве. Алгоритм Прима.

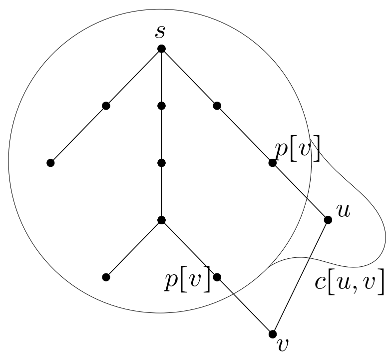
Примечание (ТГ постановка). Задан связный неориентированный граф G, неотрицательная весовая функция $c: E \to \mathbb{R}_+$.

Требуется найти связный остовной подграф графа G минимального веса.

Замечание. Существует связный остовной подграф минимального веса, который является остовным деревом.

Примечание (Алгоритм Прима). Взвешенный неориентированный граф G будет представлен весовой матрицей, то есть симметричной матрицей $C = (c_{ij})$ размера $n \times n$, где

$$c_{uv} = \begin{cases} \text{вес ребра } uv, & \text{если } uv \in E, \\ \infty, & \text{если } uv \notin E. \end{cases}$$



$$\begin{array}{ll} \text{Было:} & d[v] = c\big[p[v], v\big] \\ \text{Стало:} & d[v] = c[u, v] \end{array}$$

22 Задача о кратчайших путях. Алгоритм Дейкстры.

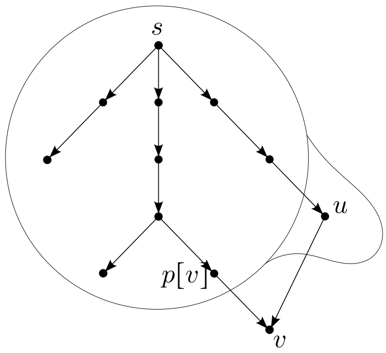
Примечание (ТГ постановка). Дан ориентированный граф G, неотрицательная весовая функция $c: E \to \mathbb{R}_+$ (вес дуг интерпретируется как их длины).

Требуется найти кратчайший путь от заданной величины $s \in V$ до заданной вершины $t \in V$ (при условии, что такой путь существует).

Замечание. Под длиной пути понимается сумма длин всех в этом пути дуг.

Примечание (Алгоритм Дейкстры). Ориентированный граф G будет представлен весовой матрицей $C = (c_{uv})$, $u, v \in V$, где

$$c_{uv} = \begin{cases} \text{вес дуги } uv, & \text{если } uv \in E, \\ \infty, & \text{если } uv \notin E. \end{cases}$$



23 Определение потока в сети. Увеличивающие пути. Лемма об увеличении потока.

Определение 49 (Поток в сети). *Потоком* из s в t в сети G называется функция $f : E \to \mathbb{R}_+$, удовлетворяющая условиям:

$$0 \leqslant f(e) \leqslant c(e) \quad \forall e \in E, \tag{3}$$

$$\sum_{v \in A(u)} f(uv) - \sum_{v \in B(u)} f(vu) = \begin{cases} b, & \text{если } u = s \\ 0, & \text{если } u \notin \{s, t\} \\ -b, & \text{если } u = t \end{cases}. \tag{4}$$

Определение 50 (Увеличивающий путь). *Увеличивающим путем* для потока f называется (s,t)-полупуть P, в котором любая прямая дуга e ненасыщена (то есть $f(e) < c(e)$), а любая обратная дуга e непуста (то есть $f(e) > 0$).

Лемма 6 (Об увеличении потока). Если для потока f в сети G существует увеличивающий путь P, то поток может быть увеличен.

24 Алгоритм Эдмондса-Карпа построения максимального потока.

Примечание (Алгоритм Эдмондса-Карпа)**.** Пусть имеется двухполюсная сеть $G = (V, E)$ (это обыкновенный ориентированный граф), $|V| = n$, $|E| = m$, и некоторый поток f из s в t в сети G.

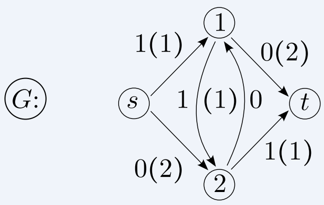
Построим вспомогательную сеть $G_f = (V, E_f)$ по правилу: $uv \in E_f \Leftrightarrow$ выполнено хотя бы одно из двух условий:

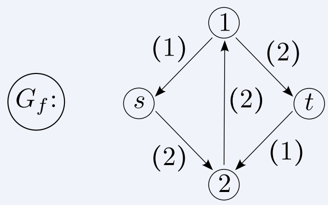
1. В G есть дуга $uv: \ f(uv) < c(uv)$.
2. В G есть дуга $vu: \ f(vu) > 0$.

Пропускные способности дуг сети G_f зададим следующим образом:

- если выполнено только условие 1., то $c_f(uv) = c(uv) - f(uv)$,
- если выполнено только условие 2., то $c_f(uv) = f(vu)$,
- если выполнены оба условия, то $c_f(uv) = c(uv) - f(uv) + f(vu)$.

Пример. $b(f) = 1$





25 Разрезы. Лемма о потоках и разрезах. Следствие.

Примечание. Рассмотрим двухполюсную сеть $G = (V, E)$ с источником s, стоком t и заданной функцией пропускных способностей $c : E \to \mathbb{R}_+$.

Определение 51 (Разрез)**.** Пусть $W \subset V$, $\widetilde{W} = V \setminus W$. *Разрезом* $(W, \widetilde{W})$ в сети G называется множество всех дуг вида $e = uv$, где $u \in W$, $v \in \widetilde{W}$.

Примечание. Говорят, что разрез разделяет вершины s и t, если $s \in W$, $t \in \widetilde{W}$.

Определение 52 (Пропускная способность)**.** *Пропускной способностью* разреза $(W, \widetilde{W})$ называется число

$$c(W, \widetilde{W}) = \sum_{e \in (W, \widetilde{W})} c(e).$$

Определение 53 (Минимальный разрез). *Минимальным разрезом*, разделяющим s и t, называется разрез с минимальной пропускной способностью среди всех таких разрезов.

Определение 54 (Поток через разрез). Если $f : E \to \mathbb{R}_+$ – поток из s в t, то потоком через разрез $(W, \widetilde{W})$ называется число

$$f(W, \widetilde{W}) = \sum_{e \in (W, \widetilde{W})} f(e).$$

Лемма 7 (О потоках и разрезах). Для любого потока f из s в t и произвольного разреза $(W, \widetilde{W})$, разделяющего s и t, имеет место равенство

$$b(f) = f(W, \widetilde{W}) - f(\widetilde{W}, W).$$

Следствие (О потоках и разрезах). В любой сети величина любого потока из s в t не превосходит пропускной способности любого разреза, разделяющего s и t.

26 Теорема Форда-Фалкерсона.

Теорема 21 (Форд-Фалкерсон). В любой конечной сети $G = (V, E)$ величина максимального потока из s в t равна пропускной способности минимального разреза, разделяющего s и t.

27 Два критерия максимальности потока.

Теорема 22 (Первый критерий). Поток f^* – макимальный $\Leftrightarrow$ не существует пути, увеличивающий f^*.

Теорема 23 (Второй критерий). Поток f^* – макимальный $\Leftrightarrow$ он насыщает все дуги некоторого разреза $(W, \widetilde{W})$, оставляя пустыми все дуги обратного разреза $(\widetilde{W}, W)$.

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

Определение 55 (Массовая задача)**.** *Массовая задача* определяется следующей информацией:

1. Общим списком всех параметров.
2. Формалировкой свойств, которыми должно обладать решение.

Определение 56 (Индивидуальная задача)**.** *Индивидуальная задача* получается из массовой подстановкой в конкретные параметры конкретных значений.

Примечание. Пусть P – массовая задача (множество всевозможных индивидуальных задач) и пусть алгоритм A решает массовую задачу P.

Определение 57 (Трудоемкость (вычислительная сложность))**.** *Трудоемкостью*, или *вычислительной сложностью*, алгоритма A называется функция

$$T_A(n),$$

которая каждому числу n ставит в соответствие максимальное время работы алгоритма по всем индивидуальным задачам I из P размера n.

Другими словами, *трудоемкость* алгоритма есть время его работы в худшем случае при решении массовой задачи размера n.

Определение 58 (Полиномиальный, экспоненциальный алгоритм)**.** Алгоритм, имеющий трудоемкость $O(n^k)$ называется *полиномиальным*.

Алгоритмы, не поддающиеся подобной оценке, называются *экспоненциальными* ($O(2^n)$, $O(n^{\log n})$, $O(n^{\sqrt{n}})$, $O(n!)$).

29 Задачи распознавания свойств. Детерминированные и недетерминированные алгоритмы. Классы P и NP. Проблема «P vs NP».

Примечание. Теория сложности вычслений строится для задач распознавания свойств. Такие задачи имеют только два возможных решения: «да» и «нет».

Определение 59 (Массовая задача распознавания P). *Массовая задача распознавания* P состоит из двух множеств:

P_I : всевозможные индивидуальные задачи.

P_Y : все индивидуальные задачи с ответом «да».

$$P = < P_I, P_Y > .$$

Определение 60 (Детерминированный алгоритм). *Детерминированный алгоритм* – получает на вход индивидуальную задачу I, затем выполняет чётко определённую последовательность действий и выдаёт решение (ответ).

Определение 61 (Недетерминированный алгоритм). *Недетерминированный алгоритм* состоит из двух стадий – стадии угадывания и стадии проверки.

На стадии угадывания по заданной индивидуальной задаче I происходит просто «угадывание» некоторой структуры S – *подсказки*.

Затем I и S вместе подаются на вход стадии проверки, которая представляет собой обычный детерминированный алгоритм и либо заканчивается ответом «да», либо ответом «нет», либо продолжается бесконечно.

Определение 62 (Класс $\mathcal{P}$). *Класс* $\mathcal{P}$ определяется как клас массовых задач распознавания, разрешимых полиномаильными алгоритмами.

Другими словами, задача распознавания P принадлежит классу $\mathcal{P}$ $\Leftrightarrow$ существует полиномиальный алгоритм, который решает задачу P.

Определение 63 (Решение задачи P недетерминированным алгоритмом). Говорят, что недетерминированный алгоритм *«решает»* массовую задачу распознавания P, если для любой индивидуальной задачи $I \in P_I$ выполнено условие: $I \in P_Y \Leftrightarrow$ существует такая подсказка S, угадывание которой для входа I приводит к тому, что стадия проверки, начиная работу на входе (I, S), заканчивается ответом «да».

Определение 64 (Полиномиальный алгоритм)**.** Недетерминированный алгоритм называется *полиномиальным*, если его стадия проверки – полиномиальный алгоритм.

Определение 65 (Класс $\mathcal{NP}$)**.** *Класс* $\mathcal{NP}$ – это класс всех массовых задач распознавания, «разрешимых» полиномиальным недетерминированными алгоритмами за полиномиальное время.

Другими словами, задача распознавания $P \in \mathcal{NP}$, если существует полнимиальный недетерминированный алгоритм, который решает.

Теорема 24. $\mathcal{P} \subseteq \mathcal{NP}$.

Теорема 25. Пусть P – множество массовых задач распознавания размера n. Если $P \in \mathcal{NP}$, то существует детерминированный алгоритм решения задачи P трудоемкости $O(2^{f(n)})$, где $f(n)$ – некоторый полином.

Примечание (Гипотеза)**.** $\mathcal{P} \neq \mathcal{NP}$.

30 Полиномиальная сводимость задач распознавания. Свойства полиномиальной сводимости. NP-полные задачи распознавания.

Примечание. В классе $\mathcal{NP}$ выделен очень большой подкласс $\mathcal{NPC}$ *«сложных» задач* (так называемые $\mathcal{NP}$*-полные задачи*).

Из $\mathcal{P} \neq \mathcal{NPC}$ следует, что из $\mathcal{NPC}$ не существует полиномиального алгоритма хотя бы для одной задачи из $\mathcal{NPC}$ сразу следовало бы, что $\mathcal{P} = \mathcal{NP}$.

Определение 66 (Полиномиальная сводимость задач)**.** Пусть P, Q – две задачи распознавания и A – такой полиномиальный алгоритм, который для любой индивидуальной задачи $I \in P_I$ строит некоторую задачу $I' = A(I) \in Q_I$.

Если при этом выполнено условие

$$I \in P_Y \Leftrightarrow I' \in Q_Y,$$

то говорят, что задача P полиномиально сводится к задаче Q,

$$P \propto Q.$$

Определение 67 ($\mathcal{NP}$-полная задача)**.** Задача распознавания Q называется $\mathcal{NP}$-полной, если:

1. $Q \in \mathcal{NP}$.
2. $\forall P \in \mathcal{NP} \quad p \propto Q$.

Класс всех $\mathcal{NP}$-полных задач обозначается

$$\mathcal{NPC}.$$

Теорема 26 (Свойства полиномиальной сходимости)**.**

1. Если $P \propto Q$ и $Q \propto R$, то $P \propto R$ (транзитивность.)
2. Пусть $P \propto Q$. Тогда $Q \in \mathcal{P} \Rightarrow P \in \mathcal{P}$.
3. Пусть $P, Q \in \mathcal{NP}$ и $P \propto Q$. Тогда $P \in \mathcal{NPC} \Rightarrow Q \in \mathcal{NPC}$.

31 NP-полные задачи распознавания. Теорема о сложности NP-полных задач. Схема доказательства NP-полноты.

Теорема 27 (О сложности $\mathcal{NP}$-полных задач)**.**

1. Если хотя бы одна $\mathcal{NP}$-полная задача полиномиально разрешима, то $\mathcal{P} = \mathcal{NP}$.
2. Если хотя бы одна задача класса $\mathcal{NP}$ труднорешаема (то есть $\mathcal{P} \neq \mathcal{NP}$), то все $\mathcal{NP}$-полные задачи труднорешаемы.

Примечание (Схема доказательства NP-полноты)**.** Да впадлу мне

32 Теорема Кука (без доказательства). Примеры NP-полных задач и сводимостей. Сведение задачи о выполнимости к задаче о клике.

Примечание (Задача о выполнимости (ВЫП))**.** *Задача о выполнимости (ВЫП)* состоит в определении, является ли данная КНФ выполнимой.

Теорема 28 (Кука)**.** Задача ВЫП $\mathcal{NP}$-полна.

Примечание (Примеры NP-полных задач и сводимостей)**.** Я устал

В монографии М. Гэри и Д. Джонсона [3] приведено более 300 $\mathcal{NP}$-полных задач, а вообще счёт идёт на тысячи.

Наиболее часто для доказательства $\mathcal{NP}$-полноты задач распознавания использовались следующие 6 задач.

3-выполнимость (*3-ВЫП*). Дана КНФ $F = D_1 \& D_2 \& \ldots \& D_p$ на конечном множестве переменных X, причём $|D_i| = 3$ для всех $i = 1, 2, \ldots, p$ (т. е. каждая дизъюнкция содержит ровно 3 литерала). Выполнима ли данная КНФ?

Трёхмерное сочетание (*3-С*). Пусть X, Y, Z — попарно непересекающиеся множества, $|X| = |Y| = |Z| = k$, задано множество $M \subseteq X \times Y \times Z$. Верно ли, что M содержит *трёхмерное сочетание*, т. е. такое подмножество $M' \subseteq M$, что $|M'| = k$ и никакие два разных элемента M' не имеют ни одной равной координаты?

Клика (*КЛИКА*). Дан граф $G = (V, E)$ и натуральное число $p \leqslant |V|$. Верно ли, что G содержит *клику* мощности, не меньшей p, т. е. такое подмножество $K \subseteq V$, что $|K| \geqslant p$ и любые две вершины из K смежны в графе G?

Вершинное покрытие (*ВП*). Дан граф $G = (V, E)$ и натуральное число $q \leqslant |V|$. Имеется ли в G *вершинное покрытие* мощности, не большей q, т. е. такое подмножество $B \subseteq V$, что $|B| \leqslant q$ и для каждого ребра $uv \in E$ хотя бы одна из вершин u, v принадлежит B?

Гамильтонов цикл (*ГЦ*). Дан граф $G = (V, E)$. Содержит ли граф G гамильтонов цикл?

Разбиение (*РАЗБ*). Дано конечное множество A и для каждого $a \in A$ задан «вес» $w(a) \in Z_+$. Существует ли такое подмножество $A' \subseteq A$, что

$$\sum_{a \in A'} w(a) = \sum_{a \in A \setminus A'} w(a)?$$

Диаграмма последовательного сведения задач

$$\begin{tikzcd}
 & & & \text{ВЫП} \arrow[dl] \arrow[dr] & & & \\
 & & \text{КЛИКА} \arrow[dl] & & \text{3-ВЫП} \arrow[dr] & & \\
 & \text{ВП} \arrow[dl] & & & & \text{3-С} \arrow[dr] & \\
\text{ГЦ} & & & & & & \text{РАЗБ}
\end{tikzcd}$$

Для примера докажем первую сводимость.