Дискретная математика
2 семестр ПИ,
Лекции

Собрано 6 апреля 2022 г. в 22:56

Содержание

Раздел #1: Кодирование информации

1.1. Задача об оптимальном префиксном коде

Пусть Λ – произвольное конечное множество (алфавит), $a \in \Lambda$ – символы. Пусть $\forall a \in \Lambda\ \exists l(a) \in \mathbb{N}, \exists c(a) = \{0,1\}^{l(a)}$ – кодовая последовательность a, где $l(a)$ – длина.
Очевидно, условие $\forall a, b \in \Lambda \to (a \neq b \Rightarrow c(a) \neq c(b))$ не является достаточным для однозначного распознавания символов.

Определение 1. Код называется префиксным, если $\forall a, b \in \Lambda\ c(a) = \omega \Rightarrow \nexists\ m \in \mathbb{N}_0 : c(b) = \omega\gamma$, где $\gamma \in \{0,1\}^m$

Пусть $\forall a \in \Lambda$ соответствует вероятность $p(a)$ появления этого символа в сообщении. $\sum_{a\in\Lambda} p(a) = 1$ и считаем $\forall a \in \Lambda\ p(a) > 0$.
Введем дискретную случайную величину $l : \forall a \in \Lambda\ Pr\{l = l(a)\} = p(a)$ – длина кодовой последовательности символа в сообщении.

Определение 2. Оптимальным называется префиксный код, минимизирующий математическое ожидание $l : \mathbb{E}l = \sum_{a\in\Lambda} l(a)p(a)$

Чем чаще встречается символ, тем короче должна быть кодовая последовательность.
Почему вообще ОПК существует? Известно, что $\mathbb{E}l \geqslant 1$ (в каждой кодовой последовательности должен быть хотя бы один символ). Всегда можно сделать префиксный код, в котором все символы имеют одинаковые длины кодовых последовательностей и эти последовательности различны ($\forall a \in \Lambda\ l(a) = \lceil \log_2(|\Lambda|) \rceil$), т.е. префиксный код существует и матожидание длины кодовой последовательности ограничено.

Лемма 1. Если в некотором коде C существует $x \in \Lambda : c(x) = \omega\alpha$, где $\alpha \in \{0,1\}$ и при этом $\nexists\ y \in \Lambda, y \neq x : c(y) = \omega\gamma$, где $\gamma \in \{0,1\}^k$ (то есть, если ω не является началом никакой другой кодовой последовательности, кроме $c(x)$), то код $C' : c'(x) = \omega, \forall y \in \Lambda, y \neq x\ c' = c(y)$ будет префиксным (по построению и условию леммы) и $\mathbb{E}l' = \mathbb{E}l - p(x)l(x) + p(x)(l(x) - 1) = \mathbb{E}l - p(x) < \mathbb{E}l$.
Тогда код C точно не мог быть оптимальным.

Лемма 2 (Лемма о кратчайшем префиксе). Если в префиксном коде C $\exists a, b \in \Lambda, a \neq b : p(a) < p(b), l(a) < l(b)$, то такой код не оптимален.

Доказательство. Проверим, что для кода C', в котором $c'(a) = c(b), c'(b) = c(a)$ и $\forall x \in \Lambda : x \neq a, x \neq b\ c'(x) = c(x)$ верно $\mathbb{E}l - \mathbb{E}l' > 0$.

$$\mathbb{E}l - \mathbb{E}l' = p(a)l(a) + p(b)l(b) - p(a)l(b) - p(b)l(a) = (p(a) - p(b))(l(a) - l(b)) > 0$$

□

Лемма 3 (Лемма о соседстве самых редких символов). Пусть $a, b \in \Lambda, a \neq b$ – символы с наименьшими вероятностями ($\forall x \in \Lambda\ p(x) \geqslant p(b) \geqslant p(a)$). Тогда $\exists$ ОПК : $c(a) = \omega 0, c(b) = \omega 1$, где $\exists k \in \mathbb{N}_0 : \omega \in \{0,1\}^k$ и это самые длинные кодовые последовательности.

Доказательство. Пусть C' – ОПК. По лемме о кратчайшем префиксе a и b имеют самые длинные кодовые последовательности в $C' : \forall x \in \Lambda, x \neq a, x \neq b\ l'(a) \geqslant l'(b) \geqslant l'(x)$
Если $c(a) = \omega\gamma, \omega \in \{0,1\}^{l'(b)}, \gamma \in \{0,1\}^{l'(a)-l'(b)}$ и ω не является началом никакой кодовой последовательности (т.к. остальные кодовые последовательности не длиннее ω и $\nexists$ символа с кодовой последовательностью ω в силу префиксности C') $\Rightarrow$ можно сократить кодовую последовательность a, создав более оптимальный код (?!).
$\Rightarrow$ из оптимальности C' следует $l(a) = l(b)$. Пусть $c'(b) = \omega 1$, тогда, если $\exists x \in \Lambda : c'(x) = \omega 0$, то построим ОПК $C : c(a) = c'(x), c(x) = c'(a), \forall z \in \Lambda, z \neq a, z \neq x\ c(z) = c'(z)$.
Если $\nexists x \in \Lambda : c'(x) = \omega 0$, то построим ОПК $C : c(a) = \omega 0, \forall z \in \Lambda, z \neq a\ c(z) = c'(z)$. □

Лемма 4 (Лемма об ОПК для расширенного алфавита). Пусть $a, b \in \Lambda, a \neq b$ – символы с наименьшими вероятностями. $\Lambda' = \Lambda \setminus \{a, b\} \cup \{\underbrace{ab}\}$, где $\underbrace{ab} \notin \Lambda$, $p(\underbrace{ab}) = p(a) + p(b)$. Пусть C' – ОПК для $\Lambda', c'(\underbrace{ab}) = \omega$. Тогда для Λ код $C : c(a) = \omega 0, c(b) = \omega 1, \forall x \in \Lambda, x \neq a, x \neq b\ c(x) = c'(x)$ будет ОПК.

Доказательство. $l(a)p(a) + l(b)p(b) = (l'(\underbrace{ab}) + 1)(p(a) + p(b)) = l'(\underbrace{ab})p(\underbrace{ab}) + p(\underbrace{ab})$.
Тогда $\mathbb{E}l = \mathbb{E}l' + p(\underbrace{ab})$.

Пусть $\overline{C}$ – ОПК для Λ и $\mathbb{E}\overline{l} < \mathbb{E}l$. По лемме о соседстве: $\overline{c}(a) = \gamma 0, \overline{c}(b) = \gamma 1$. Построим $\overline{C}'$ для $\Lambda' : \overline{c}'(\underbrace{ab}) = \gamma$ и $\forall x \in \Lambda, x \neq a, x \neq b\ \overline{c}'(x) = \overline{c}(x)$.

$\overline{C}'$ – префиксный? По Лемме о кратчайшем префиксе $\nexists$ символа с кодовой последовательностью длины $> \overline{l}(a)$. Никакой символ не мог иметь кодовую последовательность γ, т.к. $\overline{C}$ префиксный. Единственные две последовательности длины $\overline{l}(a)$, начинающиеся на γ – это коды a и b. Но их нет в Λ'. При этом $\mathbb{E}\overline{l} = \mathbb{E}\overline{l}' = p(\underbrace{ab})$. По предположению $\mathbb{E}l' + p(\underbrace{ab}) = \mathbb{E}l > \mathbb{E}\overline{l} = \mathbb{E}\overline{l}' + p(\underbrace{ab})$ (?!) оптимальности $C' \Rightarrow \mathbb{E}\overline{l} \geqslant \mathbb{E}l$, но т.к. $\overline{C}$ – ОПК $\Rightarrow \mathbb{E}\overline{l} = \mathbb{E}l$ и C – ОПК. □

Задача: нужно построить ОПК на алфавите $\Lambda, |\Lambda| = M$. По лемме об ОПК для расширенного алфавита задачу построения ОПК можно свести к такой же задаче, но с исходным алфавитом с числом букв на единицу меньше, и с набором вероятностей, получающимся из первоначального сложением двух наименьших вероятностей.
Уменьшаем пока не получится алфавит из двух букв. ОПК для алфавита из 2-х букв – $\{0,1\}$.
Строже: $\Lambda_0 := \Lambda$. $\forall k \in 0...(M-3)$ берем $a_k, b_k \in \Lambda_k : \forall x \in \Lambda_k, x \neq a_k, x \neq b_k\ p(a_k) \leqslant p(b_k) \leqslant p(x)$ и построим $\Lambda_{k+1} = \Lambda_k \setminus \{a_k, b_k\} \cup \{\underbrace{a_k b_k}\}$...

Для $\Lambda_{M-2} = \{a_{M-2}, b_{M-2}\}$ оптимальным будет код $C_{M-2} : c_{M-2}(a_{M-2}) = 0, c_{M-2}(b_{M-2}) = 1$, т.к. для него $\mathbb{E}l_{M-2} = 1$.
Теперь для $k \in 1...(M-2)$ есть ОПК C_k для Λ_k. По лемме об ОПК для расширенного алфавита

строится ОПК C_{k-1} для Λ_{k-1} такой, что $c_{k-1}(a_{k-1}) = c_k(a_{k-1}b_{k-1})0, c_{k-1}(b_{k-1}) = c_k(a_{k-1}b_{k-1})1, \forall x \in \Lambda_k, x \neq \underbrace{a_{k-1}b_{k-1}}\ c_{k-1}(x) = c_k(x)$.

Выполняем, пока не получится C_0 – ОПК для $\Lambda_0 = \Lambda$.

Пример. $\Lambda_0 = \{a,b,c,d,e,f,g\}, p(a) = 0.13, p(b) = 0.08, p(c) = 0.25, p(d) = 0.18, p(e) = 0.03, p(f) = 0.12, p(g) = 0.21$.
$a_0, = e, b_0 = b, \Lambda_1 = \{a, \underbrace{e,b}, c,d,f,g\}, p(a) = 0.13, p(\underbrace{eb}) = 0.11, p(c) = 0.25, p(d) = 0.18, p(f) = 0.12, p(g) = 0.21$.
$a_1 = \underbrace{eb}, b_1 = f, \Lambda_2 = \{a, \underbrace{ebf}, c,d,g\}, p(a) = 0.13, p(\underbrace{ebf}) = 0.23, p(c) = 0.25, p(d) = 0.18, p(g) = 0.21$.
$a_2 = a, b_2 = d, \Lambda_3 = \{\underbrace{ad}, \underbrace{ebf}, c, g\}, p(\underbrace{ad}) = 0.31, p(\underbrace{ebf}) = 0.23, p(c) = 0.25, p(g) = 0.21$.
$a_3 = g, b_3 = \underbrace{ebf}, \Lambda_4 = \{\underbrace{ad}, \underbrace{gebf}, c\}, p(\underbrace{ad}) = 0.31, p(\underbrace{gebf}) = 0.44, p(c) = 0.25$. $a_4 = c, b_4 = \underbrace{ad}, \Lambda_5\{\underbrace{cad}, \underbrace{gebf}\}, p(\underbrace{cad}) = 0.56, p(\underbrace{gebf}) = 0.44$. Тогда $c_5(\underbrace{gebf}) = 0, c(\underbrace{cad}) = 1$.
Теперь раскрываем алфавит обратно:
$c_4(\underbrace{gebf}) = 0, c_4(c) = 10, c_4(\underbrace{ad}) = 11$.
$c_3(g) = 00, c_3(\underbrace{ebf}) = 01, c_3(c) = 10, c_3(\underbrace{ad}) = 11$.
$c_1(g) = 00, c_1(\underbrace{eb}) = 010, c_1(f) = 011, c_1(c) = 10, c_1(a) = 110, c_1(d) = 111$.
$c_0(g) = 00, c_0(e) = 0100, c_0(b) = 0101, c_0(f) = 011, c_0(c) = 10, c_0(a) = 110, c_0(d) = 111$.

1.2. Неравенство Крафта

Пусть задан набор длин $l_1, ..., l_m$, не все обязательно различны. Может ли такой набор оказаться набором длин некоторого префиксного кода.

Теорема 1. Для того, чтобы набор длин $l_1, ..., l_m$ мог быть набором длин кодовых последовательностей некоторого ПК для алфавита из m символов необходимо и достаточно, чтобы $\sum_{i=1}^{m} 2^{-l_i} \leqslant 1$.

Доказательство. " $\Rightarrow$ ". Пусть $\exists$ префиксный код для алфавита с кодовыми последовательностями с длинами $l_1, ..., l_m$. множество кодовых последовательностей – набор всех путей на двоичном дереве от корня к листьям.
Корень – нулевой уровень. Далее последовательно увеличиваем номер по мере удаления от корня.
Каждой вершине v на уровне t сопоставим число $a(v) = 2^{-t}$
Пусть вершина v на уровне t – не лист. Т.е. на уровне $t+1$ есть $\geqslant 1$ вершина, получившаяся из v. Обозначим её $N(v)$. Тогда $a(v) \geqslant \sum\limits_{u \in N(v)} a(u)$

Просуммируем неравенства для всех не листов:

$$\sum_{v \text{ не лист}} a(v) \geqslant \sum_{u \text{ не корень}} a(u)$$

$\Rightarrow 2^0 \geqslant \sum_{u \text{ листья}} a(u)$. Необходимость доказана.
" $\Leftarrow$ ". Считаем, что выполнено неравенство и пусть $l_1 \leqslant l_2 \leqslant ... \leqslant l_m$
n_j– число листьев на уровне $j : n_j = |\{i : l_i = j, i \in 1 : m\}|$

$\sum_{i \in 1:m} 2^{-l_i} \geqslant 1 \Rightarrow \sum_{j \in 1:l_m} 2^{-j} n_j \leqslant$. Тогда $\forall j \in 1 : l_m : n_j \leqslant 2^j - (2^{j-1}n_1 + ... + 2n_{j-1})$
Пусть $m \neq 1$. Выделим на первом уровне вершин $n_1 \leqslant 2$, на втором уровне останется $2(2 - n_1)$. Известно, что $n_2 \leqslant 2^2 - 2n_1 \Rightarrow$ осталось не меньше, чем требуется для второго уровня.
$(j-1)$–уровень: было свободно $2^{j-1} - (2^{j-2}n_1 + ... + 2n_{j-2})$ и n_{j-1} не больше этой величины. Выделим n_{j-1} узлов, останется $2^{j-1} - (2^{j-2}n_1 + ... + 2n_{j-2}) - n_{j-1}$. Значит на j-м уровне будет $2 \cdot (...) = 2^j - (2^{j-1}n_1 + ... + 2n_{j-1})$ □

1.3. Напоминалка

Пусть S – конечное множество. $|S| = n$.
Пусть задана функция $f : S \to [0,1], \forall \omega \in S \exists! f(\omega) \in [0,1]$
$\sum_{\omega \in S} f(\omega) = 1$. Определим $\forall A \subseteq S$ величину $Pr(A) = \sum_{\omega \in A} f(\omega)$
Функция f в общем-то и не нужна. Достаточно иметь Pr.

Определение 3. (S, Pr) называется вероятностным пространством.
S – пространство элементарных событий.
$\omega \in S$ – элементарное событие (исход). $A \subseteq S$ – событие. Pr(A) – вероятность A.
$A, B \subseteq S, Pr(A \cap B) = 0$ – несовместные события.

Свойства вероятности:

- $Pr(A \cup B) = Pr(A) + Pr(B) - Pr(A \cap B)$
- $Pr(A) + Pr(S \backslash A) = 1$
- $Pr(A \cup B) \leqslant Pr(A) + Pr(B)$
- $Pr(A) = Pr(A \backslash B) + Pr(A \cap B)$

Неравенство Йенсена:

Определение 4. Функция f называется выпуклой на $X \in R$, если $\forall x_1, x_2 \in X$ и $\forall \alpha \in [0,1]$ выполняется неравенство $f(\alpha x_1 + (1 - \alpha x_2) \leqslant \alpha f(x_1) + (1 - \alpha) f(x_2)$

Неравенство Йенсена: пусть f выпуклая на X функция. Тогда $f(\sum_{i=1}^{n} \alpha_i x_i) \leqslant \sum_{i=1}^{n} \alpha_i f(x_i)$, где

$x_i \in X, \alpha_i \geqslant 0, \sum_{i=1}^{n} \alpha_i = 1$.

Доказательство. База при n = 2 верна по определению выпуклой функции. Пусть f – выпуклая на X функция. Тогда

$$f(\sum_{i=1}^{n+1}\alpha_i x_i) = f((1-\alpha_{n+1})\sum_{i=1}^{n}\frac{\alpha_i}{1-\alpha_{n+1}}x_i + \alpha_{n+1}x_{n+1}) \leqslant (1-\alpha_{n+1})f(\sum_{i=1}^{n}\frac{\alpha_i}{1-\alpha_{n+1}}x_i) + \alpha_{n+1}f(x_{n+1}) \leqslant$$

$$\leqslant (1-\alpha_{n+1})\sum_{i=1}^{n}\frac{\alpha_i}{1-\alpha_{n+1}}f(x_i) + \alpha_{n+1}f(x_{n+1}) = \sum_{i=1}^{n+1}\alpha_i f(x_i)$$ □

1.4. Конечная случайная схема

Определение 5. Пусть $A_1, A_2, ..., A_n$– разбиение множества исходов S вероятностного пространства (S, Pr). Конечной случайной схемой называется схема α, сопоставляющая каждому A_i вероятность $Pr(A_i)$

Определение 6. Энтропией КСС называется $H(\alpha) = -\sum_{i=1}^{n} Pr(A_i) \times \log Pr(A_i)$

Свойства энтропии:

- $H(\alpha) \geqslant 0$
- Энтропия характеризует неопределенность, заключенную в КСС
- Для любой $\alpha \subset k$ исходами справедливо $H(\alpha) \leqslant \log k$

Доказательство. $f(x) := -x \cdot \log x$. На $[0,1]$ функция $f(x)$ строго вогнутая ⇒ по неравенству Йенсена $\sum_{i=1}^{n}\lambda_i \cdot f(x_i) \leqslant f(\sum_{i=1}^{n}\lambda_i \cdot x_i)$, причём равенство ⇔ $x_1 = ... = x_n$.

Тогда возьмём $x_i = Pr(A_i)$ и $\lambda_i = \frac{1}{k}$ $\forall i \in 1...k$, получаем $\sum_{i=1}^{k}\frac{1}{k}(-Pr(A_i) \times \log Pr(A_i)) \leqslant$

$$\leqslant -\sum_{i=1}^{k}\frac{1}{k}Pr(A_i) \times \log(\sum_{i=1}^{k}Pr(A_i))$$

$$-\frac{1}{k}\sum_{i=1}^{k}Pr(A_i) \times \log Pr(A_i) \leqslant -\frac{1}{k}\log\frac{1}{k}$$

$$-\sum_{i=1}^{k}Pr(A_i) \times \log Pr(A_i) \leqslant \log k$$

Максимальная энтропия для КСС имеет схема с k равновероятностными исходами.
$H(\alpha) = 0 \Leftrightarrow \exists!$ достоверный исход в α □

Пусть есть КСС α с исходами $A_1, ..., A_k$ и КСС β с исходами $B_1, ..., B_l$. Их пересечением $\alpha \cap \beta$ называются КСС, исходы которой – $A_i \cap B_j, \forall i \in 1, ..., k, j \in 1, ..., l$

Тогда $H(\alpha \cap \beta) = -\sum_{i=1}^{k}\sum_{j=1}^{l}Pr(A_i \cap B_j) \times \log Pr(A_i \cap B_j)$

Т.к. $Pr(A_i \cap B_j) = Pr(A_i) \times Pr(B_j|A_i) \Rightarrow H(\alpha \cap \beta) =$

$$= -\sum_{i=1}^{k}\sum_{j=1}^{l} Pr(A_i)Pr(B_j|A_i) \times (\log Pr(A_i) + \log Pr(B_j|A_i)) =$$
$$= -\sum_{i=1}^{k}\sum_{j=1}^{l} Pr(A_i)Pr(B_j|A_i) \times \log Pr(A_i) - \sum_{i=1}^{k}\sum_{j=1}^{l} Pr(A_i)Pr(B_j|A_i) \times \log Pr(B_j|A_i) =$$
$$= -\sum_{i=1}^{k} Pr(A_i) \cdot \log Pr(A_i) \cdot \sum_{j=1}^{l} Pr(B_j|A_i) + \sum_{i=1}^{k} Pr(A_i) \cdot (-\sum_{j=1}^{l} Pr(B_j|A_i) \cdot \log Pr(B_j|A_i)) =$$
$$= -\sum_{i=1}^{k} Pr(A_i) \cdot \log Pr(A_i) + ... = H(\alpha) + ...$$

Определение 7. Величину $H(\beta|A_i) := -\sum_{j=1}^{l} Pr(A_i)Pr(B_j|A_i) \cdot \log Pr(B_j|A_i)$ называют условной энтропией β при условии A_i

Определение 8. Величину $H_\alpha(\beta) := \sum_{i=1}^{k} Pr(A_i) \cdot H(\beta|A_i)$ называют средней условной энтропией β при условии α.

Таким образом, $H(\alpha \cap \beta) = H(\alpha) + H_\alpha(\beta)$
Докажем, что $0 \leqslant H_\alpha(\beta) \leqslant H(\beta)$
Неотрицательность следует из неотрицательности энтропий.
fix j, $f(x) = -x \cdot \log x, \lambda_i = Pr(A_i), x_i = Pr(B_j|A_i) \quad \forall i \in 1...k$
Неравенство Йенсена: $\sum_{i=1}^{k} Pr(A_i) \cdot (-Pr(B_j|A_i) \cdot \log Pr(A_i)) \leqslant$

$$\leqslant (-\sum_{i=1}^{k} Pr(A_i) \cdot Pr(B_j|A_i)) \cdot \log \sum_{i=1}^{k} Pr(A_i) \cdot Pr(B_j|A_i)$$

ПЧ $= (-\sum_{i=1}^{k} Pr(A_i) \cdot Pr(B_j|A_i)) \cdot \log \sum_{i=1}^{k} Pr(A_i) \cdot Pr(B_j|A_i) =$

$$= -(\sum_{i=1}^{k} Pr(B_j \cap A_i)) \cdot \log \sum_{i=1}^{k} Pr(B_j \cap A_i) = -Pr(B_j) \cdot \log Pr(B_j)$$

Просуммируем по j: $\sum_{j=1}^{j}\sum_{i=1}^{k} Pr(A_i) \cdot (-Pr(B_j|A_i) \cdot \log Pr(A_i)) \leqslant \sum_{j=1}^{l}(-Pr(B_j) \cdot \log Pr(B_j))$

$$\sum_{i=1}^{k} Pr(A_i) \cdot \sum_{j=1}^{l}(-Pr(B_j|A_i) \cdot \log Pr(A_i)) \leqslant -\sum_{j=1}^{l} Pr(B_j) \cdot \log Pr(B_j)$$

$$\sum_{i=1}^{k} Pr(A_i) \cdot H(\beta|A_i) \leqslant H(\beta) \Rightarrow H_\alpha(\beta) \leqslant H(\beta)$$

$H_\alpha(\beta) = H(\beta) \Leftrightarrow$ все $Pr(B_j|A_i)$ равны между собой.
Формула полной вероятности: $\forall j \in 1...l Pr(B_j) = \sum_{i=1}^{k} Pr(B_j|A_i) \cdot Pr(A_i)$

$$\forall j \in 1...l Pr(B_j) = Pr(B_j|A_1) \cdot \sum_{i=1}^{k} Pr(A_i) = Pr(B_j|A_1).$$

То есть $\forall i \in 1...k, j \in 1...l \quad Pr(B_j) = Pr(B_j|A_i)$

Определение 9. События A и B – взаимно независимы $\Leftrightarrow Pr(A \cap B) = Pr(A) \cdot Pr(B) \Leftrightarrow$

$Pr(A) \cdot Pr(B|A) = Pr(A) \cdot Pr(B) \Leftrightarrow Pr(B|A) = Pr(B)$

Определение 10. КСС α и β называются независимыми, когда все исходы α независимы со всеми исходами β. В таком случае $H_\alpha(\beta)$ максимальна и равна $H(\beta)$

1.5. Количество информации

Определение 11. Величина $I(\alpha, \beta) = H(\beta) - H_\alpha(\beta)$ называется количеством информации.

Свойства:

1. $I(\alpha, \beta) \geqslant 0$
2. $I(\alpha, \beta) = H(\beta) \Leftrightarrow H_\alpha(\beta) = 0$
3. $I(\alpha, \beta) = I(\beta, \alpha)$
4. $I(\alpha, \beta) = 0 \Leftrightarrow \alpha$ и β независимы.

Пример. Загадано натуральное число $x \in 1...N$
β– опыт, состоящий в нахождении x, β_m – опыт, показывающий, делится ли x на m, $m \in 1...N$.
У β есть N исходов, у β_m – два исхода.
$H_{\beta_m}(\beta) = Pr(x \vdots m) \cdot H(\beta|x \vdots m) + Pr(x \nmid m) \cdot H(\beta|x \nmid m)$
$q := \left|\frac{N}{m}\right|$ – количество чисел от 1 до N, делящихся на m. Тогда $Pr(x \vdots m) = \frac{q}{N}, Pr(x \nmid m) = \frac{N-q}{N}$.
$H(\beta|x \vdots m) = -\sum\limits_{i \vdots m, i \in 1...m} \frac{1}{q} \cdot \log \frac{1}{q} = -\frac{q}{q} \cdot \log \frac{1}{q} = \log q.$
Аналогично $H(\beta|x \nmid m) = \log(N-q) \Rightarrow H_{\beta_m}(\beta) = \frac{q}{N} \cdot \log q + \frac{N-q}{N} \cdot \log(N-q)$
$I(\beta_m, \beta) = \log N - \frac{q}{N} \cdot \log q - \frac{N-q}{N} \cdot \log(N-q) =$
$= \frac{q}{N} \cdot \log N - \frac{q}{N} \log q - \frac{N-q}{N} \cdot \log N - \frac{N-q}{N} \cdot \log(N-q) =$
$= -\frac{q}{N} \cdot \log \frac{q}{N} - \frac{N-q}{N} \cdot \log \frac{N-q}{N} \leqslant \log 2$
Равенство достигается при $q = N - q = \frac{N}{2}$.

Пример (Данетки). Загадано число от 1 до N.
Опыт β – угадать число.
Опыт α – задать любой общий (да/нет) вопрос и получить ответ.
$H(\beta) = \log N$ (Числа загаданы с равной вероятностью)
$H(\alpha) \leqslant \log 2$ (Поскольку есть всего 2 варианта ответа)
$H(\alpha_1\alpha_2...\alpha_k) \leqslant \log 2^k = k \log 2$ (k вопросов, 2 варианта ответа)
Чтобы угадать число потребуется $k \geqslant \frac{\log N}{\log 2} = \log_2 N$ вопросов.

Есть ли алгоритм, который умеет угадывать загаданное число за $O(\log N)$.

1.5.1. Избыточное кодирование

Есть сообщение $u \in \{0,1\}^k$, которое нужно передать.
Можем передавать сообщение $x(u) \in \{0,1\}^n, n \geqslant k$, содержащую некоторую избыточную информацию (канал связи шумит и может допускать ошибки), но не более d ошибок на сообщение.
β заключается в нахождении всех d ошибок. Сколько у β исходов? Для каждого количества ошибок j от 0 до d есть $\binom{n}{j}$ вариантов их расположения, то есть всего исходов у β ровно $\sum_{j=0}^{d}\binom{n}{j}$

Следовательно, $H(\beta) = \log\sum_{j=0}^{d}\binom{n}{j}$
α – дополнительное сообщение размера $n-k$. Их 2^{n-k} и $\Rightarrow H(\alpha) = \log 2^{n-k} = (n-k)\log 2$
Чтобы гарантированно найти все ошибки нужно $H(\alpha) \geqslant H(\beta)$

$$(n-k)\log 2 \geqslant \log\sum_{j=0}^{d}\binom{n}{j} \Rightarrow n-k \geqslant \log_2\sum_{j=0}^{d}\binom{n}{j} \Rightarrow k \leqslant n - \log_2\sum_{j=0}^{d}\binom{n}{j}$$

Таким образом, если канал связи допускает не более d ошибок, то для передачи сообщения размером k понадобится не менее $k + \log_2\sum_{j=0}^{d}\binom{n}{j}$.
Или, поскольку количество ошибок обычно зависит от размера переданного сообщения, если передаётся n бит и из них не более d могут быть ошибочными, то в переданном сообщении можно закодировать сообщение длиной не более $n - \log_2\sum_{j=0}^{d}\binom{n}{j}$.

1.5.2. Код Хэмминга

Предыдущая задача при $d = 1$. Известно, что $2^{n-k} \geqslant \sum_{j=0}^{1}\binom{n}{j} = 1+n$.
$l := n-k-$ длина "избыточного сообщения". Тогда $k \leqslant 2^l - l - 1$.
Чем большее сообщение, тем относительно меньше лишней информации. Как передавать дополнительную информацию?

Пример. Пусть k = 12 и мы хотим передать сообщение u = 101101011100. Зарезервируем в сообщении длины 17 места с номерами $2^i(1,2,4,8,16)$, а на остальные позиции запишем сообщение:
$x_0(u)$ = __1_011_0101110_0
Подберём на позицию 2^i такую цифру, чтобы произведение $x(u)$ и i-й строки матрицы было равно 0. На "неопределенных"позициях в строке с номером i стоят $0(2^j = 10\ldots0)$.
На позиции 2^i в i–й строке стоит 1.
?∗1+_∗0+1∗1+_∗0+0∗1+1∗0+1∗1+_∗0+0∗1+1∗0+0∗1+1∗0+1∗1+1∗0+0∗1+_∗0+0∗1 =
=? + 1 + 1 + 1 = 1+? = 0 ⇒? = 1.
Получается x_1 = 1_1_011_0101110_0
Аналогично делаем для остальных. Итого $x(u)$ = 11110110010111000
Как определять позицию ошибки? y = 11110110000111000
Посчитаем $A \times y^T = (0,1,0,1,0)^T$ – двоичная запись позиции с ошибкой.

Старший бит – справа. Почему так?

При умножении на i -ю строку матрицы j -я позиция сообщения влияла только если $A[i, j] = 1$, то есть если на i-м месте в двоичной записи числа j стояла 1 $\Rightarrow$ результат произведения строки матрицы на столбец сообщения изменился (став 1) только для тех строк, где на 10-й позиции стояла 1 (а это строки с номерами, равными позициям, где в двоичной записи числа 10 стоят 1), а для остальных строк остался 0.

Раздел #2: Графы

Определение 12. Ориентированным графом называют $G = (V, E)$, где $V \neq \varnothing$ – множество вершин, $E \subseteq V \times V$ – множество ребер. Ребра часто записывают по их концам: (v_1, v_2) или v_1v_2.

Замечание. $V = \varnothing$ иногда встречается в доказательствах утверждений. Пустым графом называют граф, множество ребер которого пусто.

Определение 13. Пусть $G = (V, E)$. $G' = (V', E')$ называют подграфом G ($G' \leqslant G$), если $V' \subseteq V, E' \subseteq (V' \times V') \cap E$. Если $E' = (V' \times V') \cap E$ подграф, то подграф называют порожденным. Порожденный граф обозначают $G[V']$.

Определение 14. Пусть $G = (V, E)$. Путем называется последовательность вершин $v_0v_1...v_n : \forall i \in 0...n \ v_i \in V, \forall i \in 1...n \ (v_{i-1}, v_i) \in E$. Простым называется путь, в котором все вершины различны.

Определение 15. Циклом называется последовательность вершин $v_0v_1...v_n : \forall i \in 0...n \ v_i \in V, \forall i \in 1...n \ (v_{i-1}, v_i) \in E, v_0 = v_n$. Простым называется цикл, в котором все вершины, кроме первой и последней, различны.

Определение 16. Ациклическим графом называется орграф без циклов.

2.1. Отношение достижимости

Определение 17. На множестве вершин V зададим отношение достижимости R^* : вершина $v_1 \in V$ находится в отношении R^* с вершиной $v_2 \in V$ (в этом случае говорят, что вершина v_2 достижима из вершины v_1), если существует с началом v_1 и концом v_2.

Замечание. Отношение достижимости для вершин орграфа рефлексивно и транзитивно, но не обязательно симметрично.

Определение 18. Определим с помощью отношения достижимости разбиение множества вершин графа на классы эквивалентности: вершины v_1, v_2 принадлежат одному классу, если отношение симметрично. Такое отношение рефлексивно, транзитивно и симметрично.

Замечание. Если граф ациклический, то каждый класс эквивалентности состоит из одной вершины.

2.1.1. Граф-покрытие и граф достижимости

Определение 19. Минимальный граф G_b, индуцирующий на множестве вершин $V(G)$ то же отношение достижимости, что и исходный орграф G (т.е. граф с неуменьшаемым далее множеством ребер), называется **базисным** графом для графа G.

Замечание. Базисный граф не обязательно единственный.

Замечание. В конечном орграфе существует базисный граф. Получается последовательным удалением ребер (v_1, v_2), для которых существует не содержащий его путь.

Определение 20. Классы эквивалентности по отношению достижимости называются связными компонентами. Классы эквивалентности по отношению взаимной достижимости называются компонентами сильной связности.

Определение 21. Пусть $G = (V, E)$ – орграф. Граф достижимости (графом транзитивного замыкания) $G^* = (V, E^*)$ для G имеет то же множество вершин V и следующее множество ребер $E^* = \{(u, v) | \text{в графе } G \text{ вершина } v \text{ достижима из вершины } u\}$

Замечание. Ребра графа достижимости G^* соответствуют путям исходного графа G.

Определение 22. Матрица смежности орграфа $G = (V, E)$ с $|V| = n$ называется матрица A_G размера $n \times n$ с элементами

$$A_{ij} = \begin{cases} 1, (v_i, v_j) \in E \\ 0 \end{cases}$$

Введем обозначения $\hat{A} := A_G \vee E_n, \hat{A}_0 = E_n, \hat{A}_1 = \hat{A}, ..., \hat{A}_{k+1} = \hat{A}_k \wedge \hat{A}$.

Лемма 5. Пусть $\hat{A}_k = \left(a_{ij}^{(k)}\right)$. Тогда

$$a_{ij}^{(k)} = \begin{cases} 1, \exists \text{ путь из } v_i \text{ в } v_j \text{ длины } \leqslant k, \\ 0 \end{cases}$$

Доказательство. Индукция по k. База верна по определению $\hat{A}_0$. Пусть верно для k, докажем для $k+1$.
$a_{ij}^{(k+1)} = a_{i1}^{(k)} a_{1j}^{(1)} \vee ... \vee a_{ir}^{(k)} a_{rj}^{(1)} \vee ... \vee a_{in}^{(k)} a_{nj}^{(1)}$. Пусть в G из v_i в v_j есть путь длины $\leqslant k+1$. Рассмотрим кратчайший из таких путей. Если длина $\leqslant k$, то $a_{ij}^{(k)} = 1$ и , т.к. $a_{jj}^{(1)} = 1$, то

$a_{ij}^{(k)} a_{jj}^{(1)} = 1$ и $a_{ij}^{(k+1)} = 1$.
Если длина ровно $k+1$, то пусть v_r – предпоследняя вершина. Тогда из v_i в v_r есть путь длины k и по предположению $a_{ir}^{(k)} = 1$. Т.к. есть ребро (v_r, v_j), то $a_{ir}^{(k)} a_{rj}^{(1)} = 1$. Поэтому $a_{ir}^{(k)} a_{rj}^{(1)} = 1$ и $a_{ij}^{(k+1)} = 1$.
В другую сторону: пусть $a_{ij}^{(k+1)} = 1$, тогда $\exists r : a_{ir}^{(k)} a_{rj}^{(1)} = 1$. Если это $r = j$, то $a_{ij}^{(k)} = 1$ и по предположению в G есть путь из v_i в v_j длины $\leqslant k$.
Если $r \neq j$, то $a_{ir}^{(k)} = 1$ и $a_{rj}^{(1)} = 1$. Это означает, что в G есть путь из v_i в v_r длины $\leqslant k$ и ребро (v_r, v_j). Объединяем и получаем путь из v_i в v_j длины $\leqslant k+1$. □

Следствие. Пусть $G = (V, E)$ – орграф, $|V| = n, G^*$ – его граф достижимости. Тогда $A_{G^*} = \hat{A}_{n-1}$.

Замечание. При вычислении можно хитрить: считать $\hat{A} \Rightarrow \hat{A}_2 \Rightarrow \hat{A}_4 \Rightarrow ...$
Также, т.к. на диагонали $\hat{A}$ стоят единицы, то $\forall i < j$ все единицы в $\hat{A}_i$ сохраняются в $\hat{A}_j$ (и в $(\hat{A}_i)^2$). При вычислении квадратов, если в "сумме"обнаруживается $r : a_{ir} = 1$ и $a_{rj} = 1$, то остальные слагаемые можно не рассматривать.

Определение 23. Граф сильной достижимости $G_*^* = (V, E_*^*)$, где $E_*^* = \{(u, v) | u, v$ взаимно достижимы в $G\}$

По матрице сильной достижимости можно выделить компоненты сильной связности графа G:

1. В первую компоненту K_1 поместить вершину v_1 и все вершины $v_j : A_{G_*^*}(1, j) = 1$
2. Построение $K_1, ..., K_i$ и v_k вершина с минимальным индексом без компоненты. Помещаем её в K_{i+1} и все $v_j : A_{G_*^*}(k, j) = 1$.

Определение 24. Пусть K и K' – компоненты сильной связности графа G. Компонента K достижима из компоненты K', если $K = K'$ или существуют две такие вершины $u \in K$ и $v \in K'$, что u достижима из v. K строго достижима из K', если $K \neq K'$ и K достижима из K'.

Определение 25. Отношение строго достижимости можно представлять в виде орграфа, вершины – компоненты сильной связности, ребра есть если есть строга достижимость – **Конденсация G, ацикличный граф**.

2.2. Турниры и полустепени в орграфе

Определение 26. Полустепень захода в орграфе для вершины – число дуг, входящих в вершину. Обозначается $d^+(v)$. Полустепень исхода в орграфе для вершины v – число дуг, исходящих из вершины $(d^-(v))$.

Определение 27. Турнир – некоторый полный орграф (V, E) (орграф без петель и между любой парой вершин есть ровно одно ребро).

Определение 28. Для ребра $(u, v) \in E$ говорим, что u доминирует над v.

Определение 29. Турнир (будучи орграфом) транзитивен, если из $(u, v) \in E, (v, w) \in E$ следует из $(u, w) \in E$.

Определение 30. Порядком турнира T называется число его вершин.

Определение 31. Полустепень выхода вершины v турнира T – число вершин, над которыми v доминирует (еще называется результатом).

Определение 32. Последовательность результатов турнира T – упорядоченная последовательность $S = (s_1, ..., s_n)$, где s_i – результат $v_i, 1 \leqslant i \leqslant n$, причем $s_1 \leqslant s_2 \leqslant ... \leqslant s_n$.

Определение 33. Множество результатов турнира T – это последовательность $D = (d_1, ..., d_m)$ различных результатов вершин турнира T, где $d_1 < d_2 < ... < d_m$.

Определение 34. Если последовательностью результатов турнира T является S, а множество результатов – D, то будем говорить, что S генерирует D.

Теорема 2 (Редеи-Камиона для пути). Любой турнир порядка n содержит гамильтонов путь (т.е. путь, содержащий все n вершин).

Упражнение. Доказательство.

Теорема 3 (Редеи-Камиона для цикла). В сильно связном турнире есть гамильтонов цикл. Верно и обратное утверждение.

Упражнение. Доказательство.

Определение 35. Вершина $v \in V(T)$ турнира T является королем $\Leftrightarrow \forall x \in V(T)$ $\exists$ путь из v в x длиной $\leqslant 2$.

Теорема 4. В любом турнире существует вершина-король.

Теорема 5. Для турнира порядка n следующие утверждения эквивалентны:

1. T транзитивен
2. T не содержит циклов длины 3
3. T ацикличен
4. Последовательность результатов турнира T – это $(0, 1, ..., n-1)$.
5. T содержит ровно один гамильтонов путь.

Доказательство. $1 \Rightarrow 2 : \exists (u,v),(v,w),(w,u)$. Но также $\exists (u,w)$ (?!)
$2 \Rightarrow 3$: $\exists$ цикл $(v_1, ..., v_k), k \geqslant 4$. Т.к. нет циклов длины 3, то есть транзитивность. Индукцией покажем, что $\exists (v_1, v_{k-1})$. База: $(v_1,v_2),(v_2,v_3) \in E \Rightarrow (v_1,v_3) \in E$. Переход: $(v_1,v_i) \in E \ \forall i < k-1$, также $(v_i,v_{i+1}) \in E \Rightarrow (v_1,v_{i+1}) \in E$. (?!) цикл (v_1, v_{k-1}, v_k).
$3 \Rightarrow 4 : D^+(T)$ – множество степеней захода. Индукция по n. База очевидна. Переход: пусть верно для $n-1$. В ациклическом графе есть вершина-сток $t : d^+(t) = 0$. Рассмотрим граф $T - t$. $D^+(T-t) = (0, 1, ..., n-2)$. А из $\forall v \in V \setminus t$ ведет одно ребро в t.
$4 \Rightarrow 5$: Существует по теореме Редеи-Камиона. Надо единственность. Докажем по индукции. База очевидна, переход: берем $s : d^-(s) = 0$ (все ребра выходят, исток). Она будет первой в гамильтоновом пути. Рассмотрим $T - s : s$ была соединена со всеми, степени уменьшились на 1 и $D^-(T-s) = (0, 1, ..., n-2)$. Значит в $T-s$ $\exists!$ гамильтонов путь. Если $\exists 2$ гамильтоновых пути с началом в s, то будет и 2 гамильтоновых пути в $T - s$ (?!).
$5 \Rightarrow 1$: $P = (v_1, ..., v_n)$ – ! гамильтонов путь. Пусть $\exists m$ – наименьший индекс : в v_m идет ребро из вершины с большим индексом, а v_k – вершина с наибольшим индексом, из которой ребро ведет в v_m.
$m \neq 1, k \neq n$: есть ребро из v_{m-1} в v_{m+1} (минимальность v_m) и из v_m в v_{m+1} (максимальность k). Есть еще цикл $P_1 = (v_1, ..., v_{m-1}, v_{m+1}, ..., v_k, v_m, v_{k+1}, ..., v_n)$.

- $m \neq 1, k = n : P_1 = (v_1, ..., v_{m-1}, v_{m+1}, ..., v_n, v_m)$
- $m = 1, k \neq n : P_1 = (v_2, ..., v_k, v_1, v_{k+1}, ...)$
- $m = 1, k = n : P_1 = (v_2, ..., v_n, v_1)$

Значит такого m не существует и $(v_i, v_j) \in E \Leftrightarrow i < j$. Значит $\forall i,j,k : 1 \leqslant i,j,k \leqslant n$ $(v_i,v_j) \in E$ и $(v_j,v_k) \in E \Rightarrow i < j \vee j < k \Rightarrow (v_i,v_k) \in E$. □

Теорема 6. Конденсация любого турнира является транзитивным турниром.

Доказательство. U, V – компоненты сильной связности. $u \in U, v \in V : (u,v) \in E$ или $(v,u) \in E$. Т.е. в конденсации есть либо ребро (U,V), либо (V,U). Рассмотрена произвольная пара вершин конденсации турнира, получилось, что она тоже турнир. Знаем, что конденсация ациклична $\Rightarrow$ транзитивна. □

Теорема 7 (Ландау). Некоторая неубывающая последовательность неотрицательных целых чисел $S = (s_1, s_2, ..., s_n)$ является последовательностью результатов некоторого турнира

$\Leftrightarrow \sum_{i=1}^{k} s_i \geqslant \frac{k(k-1)}{2}, 1 \leqslant k \leqslant n$, причем равенство при $k = n$.

Замечание. Восстановление турнира по некоторому допустимому множество результатов – это более сложная задача, чем восстановление турнира по некоторой допустимой последовательности результатов.

Теорема 8 (Яо). Если $m \geqslant 1, D = (d_1, ..., d_m)$ – множество неотрицательных чисел, то существует турнир с множеством результатов D.

Замечание. Теорема Яо доказывает только существование соответствующего турнира, но не дает способ его построения.

Замечание. Проверка существования турнира с заданной последовательностью результатов – линейная задача (Ландау). Построение турниров по последовательности результатов делается быстро (квадратичные алгоритмы).

Замечание. А как построить турнир по множеству? Строят по множеству последовательность (за полиномиальное время), а дальше понятно.

2.3. Деревья

Обозначение. $2_V := \{\{u, v\} : u, v \in V\}$

Определение 36. Неориентированным графом называют $G = (V, E)$, где $V \neq \varnothing$ – множество вершин, $E \subseteq 2_V$ – множество ребер.

Определение 37. Вершины $v_1, v_2 \in V$ называются смежными, если $v_1v_2 \in E$.

Определение 38. Вершины v_1 и v_2 графа G называются связанными, если в графе существует путь между ними. Граф называется связным, если любые две его вершины связаны. Очевидно, связанность вершин – отношение эквивалентности, и все вершины графа по этому отношению разбиваются на классы эквивалентности – множества попарно связанных вершин. Эти классы будем называются компонентами связности графа G. Компонентами графа G будем называть подграфы, индуцированные на его компонентах связности.

Замечание. Компонента связности – максимальное по включению связное множество вершин графа. Часто под компонентами связности графа G подразумевают максимальные связные подграфы этого графа, которые мы будем называть просто компонентами.

Теорема 9 (Эквивалентные определения дерева). 1. Связный граф без циклов
2. Граф, в котором $\forall v_1, v_2 \in V$ $\exists!$ простой путь между ними.
3. Связный граф, в котором $\forall v_1, v_2 \in V$ не более одного простого пути между ними.
4. Граф без циклов, в котором $\forall v_1, v_2 \in V$ $\exists$ путь.
5. Связный граф, в котором $|V| = |E| + 1$.
6. Граф без циклов, в котором $|V| = |E| + 1$.
7. Граф, в котором $|V| = |E| + 1$, $\forall v_1, v_2 \in V$ не более одного простого пути между ними.

Теорема 10 (Эквивалентные определения дерева – 2). 1. T – дерево.
2. T – минимальный связный граф.
3. T – максимальный граф без циклов.

Определение 39. Произвольный граф без циклов называются **лесом**.

Определение 40. Подграф $H \leqslant G$ называются остовом G, если $V(H) = V(G)$.

Утверждение 1. У каждого графа существует остовный лес (а у связного графа – остовное дерево).

Доказательство. Покажем для связного графа. Если в графе есть цикл, то можно удалить из этого цикла ребро. Граф, очевидно, останется связным. Продолжим такие действия до тех пор, пока циклы не исчезнут (с каждым шагом уменьшается количество ребер, изначально оно конечно). В результате получим связный граф без циклов, являющийся остовным подграфом исходного графа, то есть, остовное дерево этого подграфа. □

Утверждение 2. $G = (V, E)$ – связный граф, то существует нумерация вершин $v_1, v_2, ..., v_n$ такая, что $\forall i \in 1 : n$ $G_i = G[\{v_1, ..., v_i\}]$ – связный.

Доказательство. База: граф из одной вершины связен по определению. Переход: пусть $G_1, ..., G_i$ построены и связны. Пусть $v \in V \setminus \{v_1, ..., v_i\}$. В силу связности G существует путь $x_0x_1...x_m$, где $x_0 = v, x_m = v_1$. Пусть $s = \max\{j \in 0 : (m-1) : x_j \notin \{v_1, ..., v_i\}\}$ Тогда положим $v_{i+1} = x_s$. Очевидно, что G_{i+1} связный. □

Утверждение 3. Пусть T – дерево. Тогда существует такая нумерация вершин $V(T) = \{v_1, ..., v_n\}$, что $\forall i \in 1 : n$ $\exists!$ $j \in 1 : (i-1)$ такое, что $v_iv_j \in E(T_i)$, где $T_i = T[\{v_1, ..., v_i\}]$.

Доказательство. Так как T связный, то существует такая нумерация, что все T_i связные. В частности, т.к. T_i связный, то существует $\geqslant$ одна вершина в $\{v_1, ..., v_{i-1}\}$, смежная с v_i. Если есть две вершины в $\{v_1, ..., v_{i-1}\}$, смежные с v_i, то т.к. T_{i-1} связный $\Rightarrow$ цикл в T (?!).

Значит такая вершина только одна. □

Утверждение 4. Связный граф на n вершинах дерево $\Leftrightarrow$ в нем $n-1$ ребро.

Доказательство. $\Rightarrow$. Следует из нумерации.
$\Leftarrow$. Граф $G : |V(G)| = n, |E(G)| = n-1$. Т.к. граф связный, то он имеет остовное дерево $T : |V(T)| = n, |E(T)| = n-1 \Rightarrow G = T \Rightarrow G$ – дерево. □

Обозначение. $w : E(G) \to R_+$ – веса ребер. Вес графа $w_G = \sum_{e \in E} w(e)$

Определение 41. Минимальным остовным деревом называется остовное дерево с минимальным весом.

Лемма 6 (о безопасном ребре). Пусть $\mathscr{T}$ – множество всех минимальных остовных деревьев связного графа G. $T \in \mathscr{T}, X \subset E(T)$. Пусть $\varnothing \neq S \subseteq V(G), Q = \{uv : u \in S, v \in V(G) \setminus S\}$, причем $X \cap Q = \varnothing$. Выберем $e \in Q : w(e) = \min_{q \in Q} w(q)$. Тогда $\exists T' \in \mathscr{T} : X \cup \{e\} \subseteq E(T')$

Доказательство. Если $e \in E(T), T' = T$. Иначе $T + e$ содержит цикл (т.к. дерево – максимальный граф без циклов). Тогда $\exists e' \in E(T) : e' \in Q$ и $e' \notin X$, т.к. $X \cap Q = \varnothing$. Т.к. $w(e) = \min_{q \in Q} w(q)$, то $w(e) \leqslant w(e')$ и $w(T') = w(T) + w(e) - w(e') \leqslant w(T) \Rightarrow$ т.к. T – минимальное остовное дерево $\Rightarrow T'$ – тоже минимальное остовное дерево. □

Замечание. Такое ребро e называется **безопасным**. Разбиение $V(G)$ на два множества $S \subseteq V(G)$ и $V(G) \setminus S$ называется **разрезом** . Множество Q из леммы – множество **пересекающих** разрез $\langle S, V \setminus S \rangle$ ребер.

Алгоритм 1 (Алгоритм Краскала). 1. Добавим все вершины G в $F : V(F) = V(G)$

2. Обходим ребра $E(G)$ в порядке неубывания весов ребер:

- Если у ребра e вершины в одной компоненты связности графа F, то добавление его в остов приведет к возникновению цикла в этой компоненте связности $\Rightarrow$ не включаем его в F
- e соединяет вершины из разных компонент связности F. Значит существует разрез $\langle S, V(F) \setminus S \rangle$: одна из компонент связности составляет одну его часть, а оставшаяся часть графа – вторую. Получается, что e – минимальное ребро, пересекающее этот разрез $\Rightarrow$ по лемме e – безопасное и его можно добавить в F.
- На последнем шаге ребро соединит две оставшиеся компоненты связности и полученный подграф будет минимальным остовным деревом графа G.

Алгоритм 2 (Прима-Ярника). Последовательное построение поддерева F для ответа для графа G. Хранится приоритетная очередь Q из вершин $G \setminus F$, ключ для вершины v – это $\min\limits_{u \in V(F), uv \in E(G)} w(uv)$ – вес минимального ребра из F в $G \setminus F$.
Также для каждой вершины хранится $u = p(v)$ – вершина, на которой достигается минимум в определении ключа. Дерево F поддерживается неявно, его ребра – пары $(v, p(v))$, где $v \in G \setminus \{r\} \setminus Q$ (r – корень F)

1. F пусто, все ключи имеют значение $+\infty$.
2. Выбирается произвольная вершина r, ее ключу присваивается значение 0.
3. На очередном шаге алгоритма (пока Q не пусто) извлекается v – минимальная вершина из Q.
   - Пробегаемся по всем ребрам $vu \in E(G)$ и, если $u \in Q$ и ее ключ $> w(v,u)$, то обновляем вершину с минимумом для u ($p(u) = v$)
   - Значение ключа равно $w(vu)$. "Релаксация"ребра vu.
   - В Q обновляем ключ для u.
   - В ответ добавляется ребро $(v, p(v))$.

2.4. Сеть

Пусть есть граф $G = (V, E)$. $\forall e \in E(G)$ $e = xy$ определим $\overrightarrow{e} = (e, x, y)$, $\overleftarrow{e} = (e, y, x)$.
$\overrightarrow{E} := \{(e, x, y) : e = xy; x, y \in V(G); e \in E(G)\}$ – заметим, что сюда входят ребра в обе стороны. Тогда $\left|\overrightarrow{E}\right| = 2\,|E(G)|$.

Для двух подмножеств множества вершин $X, Y \subseteq V(G)$ определим множество $\overrightarrow{E}(X,Y) := \left\{(e, x, y) : x \in X; y \in Y; (e, x, y) \in \overrightarrow{E}\right\}$.

Для любой функции $f : \overrightarrow{E} \to \mathbb{R}$ определим $\forall X, Y \subseteq V(G)$

$$f(X,Y) = \sum_{\overrightarrow{e} \in \overrightarrow{E}(X,Y)} f\left(\overrightarrow{e}\right)$$

Определение 42. Пусть имеется произвольный неорграф G.
Вершины $s, t \in V(G), s \neq t$ назовем истоком (source) и стоком (sink), если любая другая вершина лежит на пути из s в t.

Определение 43. Функция $c \in \overrightarrow{E} \to \mathbb{N}_0$ на графе G – пропускные способности ребер.

Определение 44. (G, s, t, c) называют **сетью**.

Определение 45. Функция $f : \overrightarrow{E} \to \mathbb{R}$ – поток (flow) в сети (G, s, t, c), если

- $\forall e \in E(G)\ f(\overrightarrow{e}) = -f(\overleftarrow{e})$ (антисимметричность, кососимметричность).
- $\forall v \in V(G), v \neq s, v \neq t\ f(\{v\}, V(G)) = 0$ (закон сохранения потока).
- $\forall \overrightarrow{e} \in \overrightarrow{E}\ f(\overrightarrow{e}) \leqslant c(\overrightarrow{e})$ (ограничение пропускной способности)

Определение 46. Разрезом (или (s,t)-разрезом) в сети (G, s, t, c) называют пару $(S, \overline{S})$, где $S \subset V(G), \overline{S} = V(G) \setminus S, s \in S, t \notin S$.

Определение 47. $f(\{s\}, V) =: |f|$ – величина потока в сети. $c(S, \overline{S}) = \sum\limits_{\overrightarrow{e} \in \overrightarrow{E}(X,Y)} c(\overrightarrow{e})$ – пропускная способность разреза.

Лемма 7 (О величине потока). $(S, \overline{S})$ – разрез в сети $G \Rightarrow f(S, \overline{S}) = |f|$

Доказательство.

$$f(S, \overline{S}) = f(S, V) - f(S, S) = f(\{s\}, V) + f(S \setminus \{s\}, V) - f(S, S)$$

Второе слагаемое обнуляется по второму свойству из определения потока, третье – по третьему (ведь для любого ребра, поток по которому мы будем прибавлять, мы будем прибавлять и поток по обратному ребру). □

Лемма 8. $(S, \overline{S})$ – разрез в сети G. Тогда $|f| \leqslant c(S, \overline{S})\ \forall f$.

Доказательство.

$$|f| = f(S, \overline{S}) = \sum_{\overrightarrow{e} \in \overrightarrow{E}(S,\overline{S})} f(\overrightarrow{e}) \leqslant \sum_{\overrightarrow{e} \in \overrightarrow{E}(S,\overline{S})} c(\overrightarrow{e}) = c(S, \overline{S})$$

□

Определение 48. Минимальным разрезом (minimal cut) называется разрез с минимально возможной пропускной способностью.

Определение 49. Остаточная пропускная способность (residual capacity) ребра $c_f(u, v) = c(u, v) - f(u, v)$. Она всегда неотрицательна из-за условия на ограничение пропускной способности.

Определение 50. Остаточная сеть – граф $G_f = G(V, E_f)$, где E_f – множество ребер с положительной остаточной пропускной способностью.

Задача о максимальном потоке (maximum flow problem): найти поток f такой, что величина потока максимальна.

Теорема 11 (Форда-Фалкерсона). Пусть в сети целые пропускные способности. Тогда величина максимального потока равна пропускной способности минимального разреза: $\max|f| = \min c(S, \overline{S})$.

Доказательство. Уже знаем, что $\forall f$, $(S, \overline{S})$ справедливо $|f| \leqslant c(S, \overline{S})$. f_0 – нулевой поток (поток на всех рёбрах равен 0). Рассмотрим следующую итеративную процедуру: пусть есть f_n – целочисленный поток и
$S_n = \{ v \in V(G) | (s = v_0, ..., v_n = v)$ – простой путь, $(e_i, v_i, v_{i+1}) = \overrightarrow{e_i} \in \overrightarrow{E}, c(\overrightarrow{e_i}) - f_n(\overrightarrow{e_i}) > 0\}$.
Пусть $t \in S_n$ (есть простой путь из истока в сток).
$\varepsilon := \min\limits_{i \in 0...n-1} (c(\overrightarrow{e_i}) - f_n(\overrightarrow{e_i}))$ и определим f_{n+1}:

- $f_{n+1}(\overrightarrow{e}) = f_n(\overrightarrow{e}) + \varepsilon$, если $\exists j \in 0...n-1 : \overrightarrow{e} = \overrightarrow{e_j}$
- $f_{n+1}(\overrightarrow{e}) = f_n(\overrightarrow{e}) - \varepsilon$, если $\exists j \in 0...n-1 : \overrightarrow{e} = \overleftarrow{e_j}$
- $f_{n+1}(\overrightarrow{e}) = f_n(\overrightarrow{e})$ иначе

Проверкой определения убеждаемся, что f_{n+1} – поток.
$|f_{n+1}| = |f_n| + \varepsilon$, т.е. на каждой итерации величина потока увеличивается на положительное целое число, а поскольку поток ограничен сверху пропускной способностью минимального разреза, алгоритм сделает конечное количество шагов.
Если $t \notin S_n$, то $(S_n, \overline{S}_n)$ – разрез, причём
$\forall \overrightarrow{e} \in \overrightarrow{E}(S_n, \overline{S}_n) : f_n(\overrightarrow{e}) = c(\overrightarrow{e}) \Rightarrow c(S_n, \overline{S}_n) = f_n(S_n, \overline{S}_n) = |f_n|$. □

Замечание. Равенство величины максимального потока и пропускной способности минимального разреза доказано конструктивно. Использовавшийся в теореме алгоритм – алгоритм Форда-Фалкерсона.

Замечание. Алгоритм работает только для целых пропускных способностей. В противном случае он может работать бесконечно долго, не сходясь к правильному ответу.

Лемма 9. Сумма потоков из источника равна сумме потоков в сток.

Лемма 10. Максимальный поток положителен тогда и только тогда, когда существует путь из источника в сток, проходящий по рёбрам с положительной пропускной способностью.

Определение 51. Увеличивающий путь – путь $(s = u_1, u_2, ..., u_k = t)$ в остаточной сети и $c_f(u_i, u_{i+1}) > 0$.

Теорема 12. Поток максимален тогда и только тогда, когда нет увеличивающего пути в остаточной сети.

Определение 52 (Общая задача линейного программирования). Найти столбец $x = x[N]$, минимизирующий $c[N] \times x[N]$, где $|N| < +\infty$, при ограничениях

$$\Omega \begin{cases} x[N_1] \geqslant \mathbb{O}[N_1], \\ A[M_1, N] \times x[N] \geqslant b[M_1], i \in M_1 \\ A[M_2, N] \times x[N] = b[M_2], i \in M_2 \\ M_1 \cap M_2 = \varnothing, \\ M = M_1 \cup M_2 \end{cases}$$

Определение 53 (Двойственная задача). Найти строку $u = u[M]$, максимизирующую $u[M] \times b[M]$ при

$$\Lambda \begin{cases} u[M_1] \geqslant \mathbb{O}[M_1], \\ u[M] \times A[M, N_1] \leqslant c[N_1], j \in N_1, \\ u[M] \times A[M, N_2] = c[N_2], j \in N_2 = N \setminus N_1. \end{cases}$$

Теорема 13 (Первая теорема двойственности). Пара двойственных задач одновременно либо имеет решение, либо нет. При этом выполняется **соотношение двойственности**

$$\inf_{x \in \Omega} f(x) = \sup_{u \in \Lambda} g(u)$$

Доказательство. Без доказательства. □

Задача: Показать двойственность связанных с величиной потока и пропускной способностью разреза экстремальных задач.

2.5. Непересекающиеся пути

Пусть $G = (V, E)$ – связный граф, s, t – две несмежные вершины.

Определение 54. Пути из s в t называются вершинно-непересекающимися, если у них нет общих рёбер.

Определение 55. Пути из s в t называются вершинно-непересекающимися, если никакие две из них не имеют общей вершины (кроме s и t).

Задачи: Может быть поставлена задача о поиске максимального количества реберно-непересекающихся путей или максимального количества вершинно-непересекающихся путей.

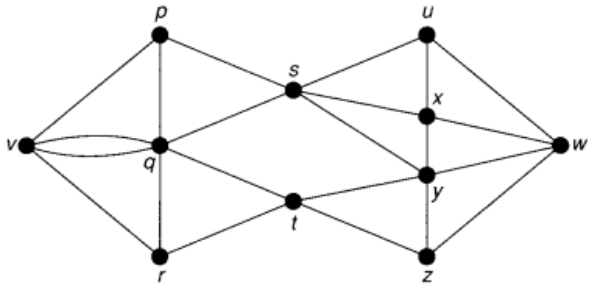

Рис. 1: Граф с 4 реберно-непересекающимися путями и 2 вершинно-непересекающимися путями

Определение 56. s,t – разделяющим множеством (s,t-disconnecting set) графа G будем называть множество $\overline{E}$ ребер $G(E)$, такое, что каждый путь от s до t включает в себя ребро из $\overline{E}$.

Определение 57. s,t – отделяющим множеством (s,t-separating set) графа G будем называть множество S вершин, отличных от s и t, таких, что каждый путь из s и t проходит через вершину из S.

Определение 58 (Альтернативные). Множество S рёбер/вершин графа G разделяет/отделяет две вершины s и t, если s и t принадлежат разным компонентам связности графа $G\backslash S$.

Замечание. Разделяющее множество рёбер мы называли разрезом.

Вернёмся к рис.1:
$E_1 = \{ps, qs, ty, tz\}$, $E_2 = \{uw, xw, yw, zw\}$ – v,w-разделяющие множества.
$V_1 = \{s,t\}, V_2 = \{p,q,y,z\}$ – v,w-отделяющие множества.
Задача: Хотим посчитать реберно-непересекающиеся пути от v в w. Если E представляет собой v,w-разделяющее множество с k ребрами, то число реберно-непересекающихся путей не может превышать k (иначе некоторое ребро из E будет включено более чем в один путь).
То есть, если E – v,w-разделяющее множество минимально возможного размера, то число реберно-непересекающихся путей равно k и в каждом таком пути имеется ровно одно ребро из E. Это, по сути, и есть реберная форма теоремы Менгера.

Теорема 14 (Менгер; Ф.-Ф., 1995 | реберная). Максимальное количество реберно-непересекающихся путей, соединяющих две различные вершины v и w связного графа, равно минимальному числу ребер в v,w-разделяющем множестве.

Доказательство. Максимальное число реберно-непересекающихся путей, соединяющих v и w, не превышает минимальное количество ребер в v,w-разделяющем множестве.
Индукцией по числу ребер в графе G покажем равенство. База очевидна. Переход: пред-

положим, что число ребер графа G равно m и что теорема верна для всех графов с менее чем m ребрами.
1) Пусть $\exists$ v, w-разделяющее множество E минимального размера k, такое, что не все его ребра инцидентны v и не все инцидентны w (E_1 из примера).
Удалим из G ребра из E, останется два непересекающихся подграфа, V и W, содержащих вершины v и w соответственно.
Определим два подграфа G_1 и G_2 из G: сожмём V (каждое его ребро) до вершины v и получим G_1; сожмём W до w и получим G_2.
Ребер в G_1 и G_2 меньше, чем в G. E – является v, w-разделяющим множеством минимального размера и для G_1, и для G_2.
По гипотезе индукции в G_1 имеется k реберно-непересекающихся путей от v до w; аналогично для G_2.
Комбинируем пути в G_1 и G_2 и получаем k реберно-непересекающихся путей в G.
2) Каждое v, w-разделяющее множество минимального размера k состоит только из ребер, которые все инцидентны v, либо все инцидентны w (множество E_2 из примера).
Можно считать, что каждое ребро графа G содержится в некотором v, w-разделяющем множестве размером k, так как в противном случае удаление соответствующего ребра не влияет на величину k и мы можем воспользоваться гипотезой индукции для получения k реберно-непересекающихся путей.
Если P – произвольный путь от v до w, то он должен состоять либо из единственного ребра, либо из двух ребер, и поэтому может содержать не более одного ребра из любого v, w-разделяющего множества размером k. Удаляя из G ребра, принадлежащие P, мы получим граф, содержащий по крайней мере $k-1$ реберно-непересекающихся путей (согласно гипотезе индукции). Вместе с P эти пути дают искомые k путей в G. □

Задача: Хотим найти число вершинно-непересекающихся путей из v в w.

Теорема 15 (Менгер, 1927 | вершинная). Максимальное число вершинно-непересекающихся путей, соединяющих две различные несмежные вершины, v и w, графа, равно минимальному числу вершин в v, w-отделяющем множестве.

Доказательство. «Рёберно-непересекающийся» и «инцидентный» → «вершинно-непересекающийся» и «смежный».
V_1 – наименьшее множество вершин, разделяющее v и w.
Разобрать три случая:

- Пусть в V_1 есть вершины, несмежные с v и несмежные с w.
- все вершины отделяющего множества V_1 смежны с v или w (пусть с v) и среди вершин V_1 есть вершина u, смежная одновременно и с v, и с w.
- все вершины V_1 смежны с v или с w (пусть с v) и среди вершин V_1 нет вершин, смежных одновременно с v и w.

□

Определение 59. Граф называется реберно k-связным (или k-реберно-связным), если удаление любых $k-1$ ребер оставляет граф связным.

Следствие. Граф G является k-реберно-связным тогда и только тогда, когда любые две различные вершины G соединяются по крайней мере k реберно-непересекающимися путями.

Определение 60. Граф G называется k-связным, если k – наибольшее из чисел, таких, что каждая пара несмежных вершин соединена не менее чем k вершинно-непересекающимися простыми путями.

Определение 61 (Альтернативное). Граф G называется вершинно k-связным (или k-связным), если удаление любых $k-1$ вершин оставляет граф связным.

Следствие. Граф G с как минимум $k+1$ вершиной является k-связным тогда и только тогда, когда любые две различные вершины G соединяются по крайней мере k вершинно-непересекающимися путями.