

Дискретная математика

2 семестр ПИ,

Лекции

Собрано 5 апреля 2022 г. в 12:29

Содержание

1. Кодирование информации	1
1.1. Задача об оптимальном префиксном коде	1
1.2. Неравенство Крафта	3
1.3. Напоминка	4
1.4. Конечная случайная схема	5
1.5. Количество информации	7
1.5.1. Избыточное кодирование	8
1.5.2. Код Хэмминга	8
2. Графы	10
2.1. Отношение достижимости	10
2.1.1. Граф-покрытие и граф достижимости	10
2.2. Турниры и полустепени в орграфе	12
2.3. Деревья	15
2.4. Сеть	18

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 $\forall a, b \in \Lambda \rightarrow (a \neq b \Rightarrow c(a) \neq c(b))$ 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 $\mathbb{E}l \geq 1$ (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 $\alpha \in \{0, 1\}^k$ 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 ($\forall x \in \Lambda p(x) \geq p(b) \geq p(a)$). 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' : $\forall x \in \Lambda, x \neq a, x \neq b l'(a) \geq l'(b) \geq l'(x)$

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 $C : c(a) = \omega 0, \forall z \in \Lambda, z \neq a c(z) = c'(z)$. \square

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

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

$\mathbb{E}l' + p(\underbrace{ab}) = \mathbb{E}l > \mathbb{E}\bar{l} = \mathbb{E}\bar{l}' + p(\underbrace{ab})$ (?) оптимальности $C' \Rightarrow \mathbb{E}\bar{l} \geq \mathbb{E}l$, но т.к. \bar{C} – ОПК

$\Rightarrow \mathbb{E}\bar{l} = \mathbb{E}l$ и C – ОПК. \square

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: $\Lambda_0 := \Lambda. \forall k \in 0 \dots (M-3)$ берем $a_k, b_k \in \Lambda_k : \forall x \in \Lambda_k, x \neq a_k, x \neq b_k p(a_k) \leq p(b_k) \leq p(x)$ и построим $\Lambda_{k+1} = \Lambda_k \setminus \{a_k, b_k\} \cup \{\underbrace{a_k b_k}\} \dots$

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 $k \in 1 \dots (M-2)$ 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 $c_{k-1}(a_{k-1}) = c_k(a_{k-1}b_{k-1})0$, $c_{k-1}(b_{k-1}) = c_k(a_{k-1}b_{k-1})1$, $\forall x \in \Lambda_k$, $x \neq \underbrace{a_{k-1}b_{k-1}}_{c_{k-1}(x)} c_{k-1}(x) = c_k(x)$.

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

Доказательство. ” \Rightarrow ”. Пусть \exists префиксный код для алфавита с кодовыми последовательностями с длинами l_1, \dots, 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 ≥ 1 вершина, получившаяся из v . Обозначим её $N(v)$. Тогда $a(v) \geq \sum_{u \in N(v)} a(u)$

Просуммируем неравенства для всех не листов:

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

$\Rightarrow 2^0 \geq \sum_{u \text{ листья}} a(u)$. Необходимость доказана.

" \Leftarrow ". Считаем, что выполнено неравенство и пусть $l_1 \leq l_2 \leq \dots \leq l_m$
 n_j – число листьев на уровне j : $n_j = |\{i : l_i = j, i \in 1 : m\}|$

$$\sum_{i \in 1 : m} 2^{-l_i} \geq 1 \Rightarrow \sum_{j \in 1 : l_m} 2^{-j} n_j \leq 1. \text{ Тогда } \forall j \in 1 : l_m : n_j \leq 2^j - (2^{j-1} n_1 + \dots + 2 n_{j-1})$$

Пусть $m \neq 1$. Выделим на первом уровне вершин $n_1 \leq 2$, на втором уровне останется $2(2 - n_1)$. Известно, что $n_2 \leq 2^2 - 2n_1 \Rightarrow$ осталось не меньше, чем требуется для второго уровня.

$(j-1)$ -уровень: было свободно $2^{j-1} - (2^{j-2} n_1 + \dots + 2 n_{j-2})$ и n_{j-1} не больше этой величины. Выделим n_{j-1} узлов, останется $2^{j-1} - (2^{j-2} n_1 + \dots + 2 n_{j-2}) - n_{j-1}$. Значит на j -м уровне будет $2 \cdot (\dots) = 2^j - (2^{j-1} n_1 + \dots + 2 n_{j-1})$ \square

1.3. Напоминка

Пусть S – конечное множество. $|S| = n$.

Пусть задана функция $f : S \rightarrow [0, 1]$, $\forall \omega \in S \exists! f(\omega) \in [0, 1]$

$\sum_{\omega \in S} f(\omega) = 1$. 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:

- $Pr(A \cup B) = Pr(A) + Pr(B) - Pr(A \cap B)$
- $Pr(A) + Pr(S \setminus A) = 1$
- $Pr(A \cup B) \leq Pr(A) + Pr(B)$
- $Pr(A) = Pr(A \setminus B) + Pr(A \cap B)$

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 $f(\alpha x_1 + (1 - \alpha) x_2) \leq \alpha f(x_1) + (1 - \alpha) f(x_2)$

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

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

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

$$\begin{aligned} f\left(\sum_{i=1}^{n+1} \alpha_i x_i\right) &= f\left((1 - \alpha_{n+1}) \sum_{i=1}^n \frac{\alpha_i}{1 - \alpha_{n+1}} x_i + \alpha_{n+1} x_{n+1}\right) \leq (1 - \alpha_{n+1}) f\left(\sum_{i=1}^n \frac{\alpha_i}{1 - \alpha_{n+1}} x_i\right) + \alpha_{n+1} f(x_{n+1}) \leq \\ &\leq (1 - \alpha_{n+1}) \sum_{i=1}^n \frac{\alpha_i}{1 - \alpha_{n+1}} f(x_i) + \alpha_{n+1} f(x_{n+1}) = \sum_{i=1}^{n+1} \alpha_i f(x_i) \quad \square \end{aligned}$$

1.4. Конечная случайная схема

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

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

Свойства энтропии:

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

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

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

$$\begin{aligned} &\leq - \sum_{i=1}^k \frac{1}{k} \text{Pr}(A_i) \times \log\left(\sum_{i=1}^k \text{Pr}(A_i)\right) \\ &= - \frac{1}{k} \sum_{i=1}^k \text{Pr}(A_i) \times \log \text{Pr}(A_i) \leq - \frac{1}{k} \log \frac{1}{k} \\ &= - \sum_{i=1}^k \text{Pr}(A_i) \times \log \text{Pr}(A_i) \leq \log k \end{aligned}$$

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

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

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

$$\begin{aligned}
&= - \sum_{i=1}^k \sum_{j=1}^l Pr(A_i) Pr(B_j|A_i) \times (\log Pr(A_i) + \log Pr(B_j|A_i)) = \\
&= - \sum_{i=1}^k \sum_{j=1}^l Pr(A_i) Pr(B_j|A_i) \times \log Pr(A_i) - \sum_{i=1}^k \sum_{j=1}^l Pr(A_i) Pr(B_j|A_i) \times \log Pr(B_j|A_i) = \\
&= - \sum_{i=1}^k Pr(A_i) \cdot \log Pr(A_i) \cdot \sum_{j=1}^l Pr(B_j|A_i) + \sum_{i=1}^k Pr(A_i) \cdot \left(- \sum_{j=1}^l Pr(B_j|A_i) \cdot \log Pr(B_j|A_i) \right) = \\
&= - \sum_{i=1}^k Pr(A_i) \cdot \log Pr(A_i) + \dots = H(\alpha) + \dots
\end{aligned}$$

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 $0 \leq H_\alpha(\beta) \leq H(\beta)$

Неотрицательность следует из неотрицательности энтропий.

fix j , $f(x) = -x \cdot \log x$, $\lambda_i = Pr(A_i)$, $x_i = Pr(B_j|A_i) \quad \forall i \in 1 \dots k$

Неравенство Йенсена: $\sum_{i=1}^k Pr(A_i) \cdot (-Pr(B_j|A_i) \cdot \log Pr(A_i)) \leq$

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

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

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

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

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

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

$H_\alpha(\beta) = H(\beta) \Leftrightarrow$ все $Pr(B_j|A_i)$ равны между собой.

Формула полной вероятности: $\forall j \in 1 \dots l Pr(B_j) = \sum_{i=1}^k Pr(B_j|A_i) \cdot Pr(A_i)$

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

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

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. $I(\alpha, \beta) \geq 0$
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 $x \in 1 \dots N$

β – опыт, состоящий в нахождении x , β_m – опыт, показывающий, делится ли x на m , $m \in 1 \dots N$.

У β есть N исходов, у β_m – два исхода.

$$H_{\beta_m}(\beta) = Pr(x:m) \cdot H(\beta|x:m) + Pr(x \nmid m) \cdot H(\beta|x \nmid m)$$

$q := \left\lfloor \frac{N}{m} \right\rfloor$ – количество чисел от 1 до N , делящихся на m . Тогда $Pr(x:m) = \frac{q}{N}$, $Pr(x \nmid m) = \frac{N-q}{N}$.

$$H(\beta|x:m) = - \sum_{i:m, i \in 1 \dots m} \frac{1}{q} \cdot \log \frac{1}{q} = -\frac{q}{q} \cdot \log \frac{1}{q} = \log q.$$

$$\text{Аналогично } H(\beta|x \nmid m) = \log(N-q) \Rightarrow H_{\beta_m}(\beta) = \frac{q}{N} \cdot \log q + \frac{N-q}{N} \cdot \log(N-q)$$

$$\begin{aligned} I(\beta_m, \beta) &= \log N - \frac{q}{N} \cdot \log q - \frac{N-q}{N} \cdot \log(N-q) = \\ &= \frac{q}{N} \cdot \log N - \frac{q}{N} \log q - \frac{N-q}{N} \cdot \log N - \frac{N-q}{N} \cdot \log(N-q) = \\ &= -\frac{q}{N} \cdot \log \frac{q}{N} - \frac{N-q}{N} \cdot \log \frac{N-q}{N} \leq \log 2 \end{aligned}$$

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.

$$H(\beta) = \log N \text{ (Числа загаданы с равной вероятностью)}$$

$$H(\alpha) \leq \log 2 \text{ (Поскольку есть всего 2 варианта ответа)}$$

$$H(\alpha_1 \alpha_2 \dots \alpha_k) \leq \log 2^k = k \log 2 \text{ (k вопросов, 2 варианта ответа)}$$

Чтобы угадать число потребуется $k \geq \frac{\log N}{\log 2} = \log_2 N$ 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 $x(u) \in \{0, 1\}^n, n \geq k$, 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 $H(\alpha) \geq H(\beta)$

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

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 $2^{n-k} \geq \sum_{j=0}^1 \binom{n}{j} = 1 + n$.

$l := n - k$ – длина "избыточного сообщения". Тогда $k \leq 2^l - l - 1$.

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 "неопределённых" позициях в строке с номером i стоят $0(2^j = 10 \dots 0)$.

На позиции 2^i в i -й строке стоит 1.

$? * 1 + _ * 0 + 1 * 1 + _ * 0 + 0 * 1 + 1 * 0 + 1 * 1 + _ * 0 + 0 * 1 + 1 * 0 + 0 * 1 + 1 * 0 + 1 * 1 + 1 * 0 + 0 * 1 + _ * 0 + 0 * 1 =$
 $= ? + 1 + 1 + 1 = 1 + ? = 0 \Rightarrow ? = 1$.

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 $V \neq \emptyset$ – множество вершин, $E \subseteq V \times V$ – множество ребер. Ребра часто записывают по их концам: (v_1, v_2) или v_1v_2 .

Замечание. $V = \emptyset$ 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 ($G' \leq G$), 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 $v_0v_1\dots v_n : \forall i \in 0\dots n \ v_i \in V, \forall i \in 1\dots n \ (v_{i-1}, v_i) \in E$. Простым называется путь, в котором все вершины различны.

Определение 15. Циклом называется последовательность вершин $v_0v_1\dots v_n : \forall i \in 0\dots n \ v_i \in V, \forall i \in 1\dots n \ (v_{i-1}, v_i) \in E, v_0 = v_n$. 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 $\hat{A}_k = (a_{ij}^{(k)})$. Тогда

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

Доказательство. Индукция по k . База верна по определению \hat{A}_0 . Пусть верно для k , докажем для $k+1$.

$a_{ij}^{(k+1)} = a_{i1}^{(k)} a_{1j}^{(1)} \vee \dots \vee a_{ir}^{(k)} a_{rj}^{(1)} \vee \dots \vee a_{in}^{(k)} a_{nj}^{(1)}$. Пусть в G из v_i в v_j есть путь длины $\leq k+1$. Рассмотрим кратчайший из таких путей. Если длина $\leq k$, то $a_{ij}^{(k)} = 1$ и, т.к. $a_{jj}^{(1)} = 1$, то

$$a_{ij}^{(k)} a_{jj}^{(1)} = 1 \text{ и } a_{ij}^{(k+1)} = 1.$$

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 $\leq k$.

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 $\leq k$ и ребро (v_r, v_j) . Объединяем и получаем путь из v_i в v_j длины $\leq k + 1$. \square

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 $\hat{A} \Rightarrow \hat{A}_2 \Rightarrow \hat{A}_4 \Rightarrow \dots$

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 $E_*^* = \{(u, v) | u, v \text{ взаимно достижимы в } G\}$

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, \dots, 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 , ациклический 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 $S = (s_1, \dots, s_n)$, где s_i – результат $v_i, 1 \leq i \leq n$, причем $s_1 \leq s_2 \leq \dots \leq s_n$.

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

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

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

Упражнение. Доказательство.

Теорема 3 (Редди-Кампона 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 ≤ 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 ацикличесен
4. Последовательность результатов турнира T – это $(0, 1, \dots, n-1)$.
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 $(v_1, \dots, v_k), k \geq 4$. 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 $T-t$. $D^+(T-t) = (0, 1, \dots, n-2)$. 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 $D^-(T-s) = (0, 1, \dots, n-2)$. Значит в $T-s$ $\exists!$ гамильтонов путь. Если $\exists 2$ гамильтоновых пути с началом в s , то будет и 2 гамильтоновых пути в $T-s$ (!).

$5 \Rightarrow 1$: $P = (v_1, \dots, v_n)$ – ! 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 $P_1 = (v_1, \dots, v_{m-1}, v_{m+1}, \dots, v_k, v_m, v_{k+1}, \dots, v_n)$.

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

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

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 ациклическа \Rightarrow транзитивна. \square

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

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

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

Теорема 8 (Яо). Если $m \geq 1, D = (d_1, \dots, d_m)$ – 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 $V \neq \emptyset$ – множество вершин, $E \subseteq 2_V$ – множество ребер.

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

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, связность вершин – отношение эквивалентности, и все вершины графа по этому отношению разбиваются на классы эквивалентности – множества попарно связанных вершин. Эти классы будем называть 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 называется **лесом**.

Определение 40. Подграф $H \leq G$ называется **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. \square

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

Доказательство. База: граф из одной вершины связан по определению. Переход: пусть G_1, \dots, G_i построены и связны. Пусть $v \in V \setminus \{v_1, \dots, v_i\}$. В силу связности G существует путь $x_0 x_1 \dots x_m$, где $x_0 = v, x_m = v_1$. Пусть $s = \max \{j \in 0 : (m-1) : x_j \notin \{v_1, \dots, v_i\}\}$ Тогда положим $v_{i+1} = x_s$. Очевидно, что G_{i+1} связный. \square

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

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

Значит такая вершина только одна. \square

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. \square

Обозначение. $w : E(G) \rightarrow R_+$ – 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 \mathcal{T} – множество всех минимальных остовных деревьев связного графа G . $T \in \mathcal{T}, X \subset E(T)$. Пусть $\emptyset \neq S \subseteq V(G), Q = \{uv : u \in S, v \in V(G) \setminus S\}$, причем $X \cap Q = \emptyset$. Выберем $e \in Q : w(e) = \min_{q \in Q} w(q)$. Тогда $\exists T' \in \mathcal{T} : X \cup \{e\} \subseteq E(T')$

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. $X \cap Q = \emptyset$. Т.к. $w(e) = \min_{q \in Q} w(q)$, то $w(e) \leq w(e')$ и $w(T') = w(T) + w(e) - w(e') \leq w(T) \Rightarrow$ т.к. T – минимальное остовное дерево $\Rightarrow T'$ – тоже минимальное остовное дерево. \square

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 компоненте 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 $\min_{u \in V(F), uv \in E(G)} w(uv)$ – 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 .
 - Пробегаясь 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 $\vec{e} = (e, x, y)$, $\overleftarrow{e} = (e, y, x)$.

$\vec{E} := \{(e, x, y) : e = xy; x, y \in V(G); e \in E(G)\}$ – заметим, что сюда входят ребра в обе стороны. Тогда $|\vec{E}| = 2|E(G)|$.

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

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

$$f(X, Y) = \sum_{\vec{e} \in \vec{E}(X, Y)} f(\vec{e})$$

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 $c \in \vec{E} \rightarrow \mathbb{N}_0$ на графе G – пропускные способности ребер.

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

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

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

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

Определение 47. $f(\{s\}, V) =: |f|$ – величина потока в сети. $c(S, \bar{S}) = \sum_{\vec{e} \in \vec{E}(S, \bar{S})} c(\vec{e})$ – пропускная способность разреза.

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

Доказательство.

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

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). \square

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

Доказательство.

$$|f| = f(S, \bar{S}) = \sum_{\vec{e} \in \vec{E}(S, \bar{S})} f(\vec{e}) \leq \sum_{\vec{e} \in \vec{E}(S, \bar{S})} c(\vec{e}) = c(S, \bar{S})$$

\square

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.