Алгоритм восстановления изображения по его коду

П. Г. Агниашвили

В рамках дискретно-геометрического подхода к распознаванию образов представлен алгоритм, вычисляющий по коду все классы a'-эквивалентных изображений с данным кодом. Алгоритм применим к изображениям произвольной размерности $n \geqslant 2$, причем его временная сложность линейно зависит от числа точек в изображении.

Ключевые слова: распознавание образов, код изображения, алгоритм восстановления изображения, временная сложность.

1. Введение

Дискретно-геометрический подход рассматривает изображения как конечные множества точек в конечномерном евклидовом пространстве. Код изображения — это определенная числовая характеристика, инвариантная относительно аффинных преобразований изображения. Данное направление получило развитие в книге [1], где, в частности, исследуется возможность восстановления изображений по их кодам в двумерном и трехмерном случаях. В работе [2] аналогичная проблема исследуется в общем случае произвольной конечной размерности. Для этого вводится вспомогательная характеристика — μ-код (модифицированный код). В отличие от основного кода μ-код содержит числа со знаком и в точности характеризует свойство изображений быть аффинно-эквивалентными с сохранением нумерации. Связь кода и μ-кода дает необходимые и достаточные условия, определяющие изображения, для которых возможно однозначное восстановление по коду.

Кроме существования такой возможности, интересна конструктивная часть вопроса — алгоритм восстановления изображения по его коду. В двумерном и трехмерном случаях такой алгоритм может быть описан на геометрическом языке. В общем случае требуется переход к аналитическому представлению кода. Основная идея заключается в расстановке знаков для элементов кода с целью получения μ-кода. Непосредственный перебор всех расстановок знаков и проверка их корректности приводит к алгоритму, временная сложность которого растет экспоненциально с ростом числа точек в изображении. С учетом того, что изображения могут содержать большое число точек, такой алгоритм не является эффективным.

В настоящей работе представлен алгоритм, вычисляющий по коду все классы изображений с данным кодом, причем временная сложность алгоритма линейно зависит от числа точек в изображении. Для этого рассматривается матричное представление кода и подматрицы специального вида. Переход от экспоненциальной к линейной сложности достигается за счет ряда закономерностей, позволяющих сократить перебор вариантов расстановок знаков. Данный алгоритм применим к изображениям произвольной размерности $n \geqslant 2$.

2. Предварительные сведения

Всюду далее рассматривается пространство $\mathbb{R}^n$, $n \geqslant 2$. Под точкой с индексом $p \in \mathbb{N}$ будем понимать упорядоченную пару $(\mathbf{x}, p)$, где $\mathbf{x} = (x_1, \ldots, x_n)$ — вектор из $\mathbb{R}^n$.

Определение 1. ([2]) *Изображением* в $\mathbb{R}^n$ называется конечное множество индексированных точек, не лежащих в одной гиперплоскости и занумерованных индексами $1, \ldots, k$ в случае k точек, $k \in \mathbb{N}$.

Для изображения A через $|A|$ будем обозначать число точек в изображении, а через $\mathbb{N}_{|A|} = \{1, \ldots, |A|\}$ — множество индексов у точек изображения.

Определение 2. ([2]) Два изображения A_1 и A_2, состоящие из одинакового числа точек ($|A_1| = |A_2|$), называются *a'-эквивалентными*, если существует аффинное преобразование, при котором каждая точка из A_1 с индексом $p \in \mathbb{N}_{|A_1|}$ отображается в точку из A_2 с тем же индексом $p \in \mathbb{N}_{|A_2|}$.

Рассмотрим изображение A. Введем произвольную аффинную систему координат в $\mathbb{R}^n$, и пусть $(x_1^p,\ldots,x_n^p)$ — координаты точки с индексом $p \in \mathbb{N}_{|A|}$ в этой системе координат. Для произвольных индексов $p_1,\ldots,p_{n+1},q_1,\ldots,q_{n+1} \in \mathbb{N}_{|A|}$ определим индексированное число $\mu_{q_1\ldots q_{n+1}}^{p_1\ldots p_{n+1}}$:

$$\mu_{q_1\ldots q_{n+1}}^{p_1\ldots p_{n+1}} = \begin{vmatrix} x_1^{p_1} & \cdots & x_n^{p_1} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{p_{n+1}} & \cdots & x_n^{p_{n+1}} & 1 \end{vmatrix} \Bigg/ \begin{vmatrix} x_1^{q_1} & \cdots & x_n^{q_1} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{q_{n+1}} & \cdots & x_n^{q_{n+1}} & 1 \end{vmatrix}$$

Здесь в случае равенства знаменателя нулю полагаем, что значение $\mu_{q_1\ldots q_{n+1}}^{p_1\ldots p_{n+1}}$ не определено, и будем обозначать такой случай знаком ∞. Для краткости обозначаем наборы индексов заглавной буквой, например, $P=(p_1,\ldots,p_{n+1})$, и пишем в таком случае $P \in \mathbb{N}_{|A|}^{n+1}$. Для индексированных чисел вида $\mu_{q_1\ldots q_{n+1}}^{p_1\ldots p_{n+1}}$ получаем короткое обозначение μ_Q^P, а наборы индексов P и Q называем *верхним* и *нижним наборами* соответственно. Перестановка индексов в верхнем или нижнем наборах может привести к изменению знака числа μ_Q^P, но не изменяет его абсолютного значения.

Определение 3. ([2]) Множество индексированных чисел $\mu\text{-}T_A = \{\mu_Q^P | P,Q \in \mathbb{N}_{|A|}^{n+1}\}$ называется *μ-кодом* изображения A. *Кодом* изображения A называется множество индексированных чисел $T_A = \{\rho_Q^P | P,Q \in \mathbb{N}_{|A|}^{n+1}\}$, где $\rho_Q^P = |\mu_Q^P|$.

Пусть для некоторого набора индексов $S=(s_1,\ldots,s_{n+1}) \in \mathbb{N}_{|A|}^{n+1}$ выполняется условие $\mu_S^P \neq \infty$. Обозначим через μ_j^p элемент μ-кода, у которого нижний набор является набором S, а верхний набор есть $(s_1,\ldots,s_{j-1},p,s_{j+1},\ldots,s_{n+1})$, то есть j-й индекс набора S замещен индексом p.

Определение 4. ([2]) Набор индексов S называется *симплексным набором*. Множество $\{\mu_j^p | p \in \mathbb{N}_{|A|}, j \in \mathbb{N}_{n+1}\}$ называется *ключевым подмножеством μ-кода* $\mu\text{-}T_A$ относительно симплексного набора S.

Аналогично случаю μ-кода вводятся понятия элемента кода, симплексного набора индексов и ключевого подмножества кода.

Лемма 1 ([2]). *Для любого индекса $p \in N_{|A|}$ выполняется соотношение:*

$$\mu_1^p + \ldots + \mu_{n+1}^p = 1. \tag{1}$$

Лемма 2 ([2]). *Для любых наборов* $P = (p_1, \ldots, p_{n+1}) \in \mathbb{N}_{|A|}^{n+1}$, $Q = (q_1, \ldots, q_{n+1}) \in \mathbb{N}_{|A|}^{n+1}$ *верна следующая формула:*

$$\mu_Q^P = \frac{\begin{vmatrix} \mu_1^{p_1} & \cdots & \mu_{n+1}^{p_1} \\ \vdots & \ddots & \vdots \\ \mu_1^{p_{n+1}} & \cdots & \mu_{n+1}^{p_{n+1}} \end{vmatrix}}{\begin{vmatrix} \mu_1^{q_1} & \cdots & \mu_{n+1}^{q_1} \\ \vdots & \ddots & \vdots \\ \mu_1^{q_{n+1}} & \cdots & \mu_{n+1}^{q_{n+1}} \end{vmatrix}} \tag{2}$$

Здесь в случае равенства знаменателя нулю полагаем $\mu_Q^P = \infty$.

Лемма 3 ([2]). *Каждому классу a'-эквивалентных изображений соответствует единственный* μ*-код, являющийся* μ*-кодом для каждого изображения из класса. Данное соответствие является биекцией.*

3. Постановка задачи

Пусть $T = (t_{ij})$ — произвольная матрица размера $k \times m$ с элементами из $\mathbb{R}$. Через $M(T)$ обозначим класс всех матриц $A = (a_{ij})$ размера $k \times m$, для которых $|a_{ij}| = |t_{ij}|$.

Определение 5. Две матрицы A и B из класса $M(T)$, называются *эквивалентными* и обозначаются $A \sim B$, если одну из них можно преобразовать в другую, умножая некоторые строки и столбцы на -1.

Пусть $\{\rho_j^p | p \in \mathbb{N}_{|A|}, j \in \mathbb{N}_{n+1}\}$ — ключевое подмножество кода T_A относительно некоторого симплексного набора индексов S, а $T^\rho = (t_{pj})$ — матрица размера $|A| \times (n+1)$, определенная соотношением $t_{pj} = \rho_j^p$, $p \in \mathbb{N}_{|A|}$, $j \in \mathbb{N}_{n+1}$.

Определение 6. Произвольная матрица $T^\mu \in M(T^\rho)$ называется *матрицей кода* T_A относительно симплексного набора индексов S. Элементы матрицы кода T^μ обозначаем через μ_j^p. Матрица кода называется *правильной*, если для каждой ее строки $(\mu_1^p, \ldots, \mu_{n+1}^p)$ выполняется соотношение (1) и для любых двух наборов индексов

$P, Q \in \mathbb{N}_{|A|}^{n+1}$ выполняется соотношение $|\mu_Q^P| = \rho_Q^P$, где число μ_Q^P определяется формулой (2).

Из леммы 2, леммы 3 и определения 6 следует простое утверждение:

Утверждение 1. *Каждому классу a'-эквивалентных изображений соответствует единственная правильная матрица кода T^μ, совпадающая с ключевым подмножеством μ-кода каждого изображения класса (при фиксированном симплексном наборе). Данное соответствие является биекцией.*

С помощью утверждения 1 задача о восстановлении изображения с точностью до a'-эквивалентности по его коду сводится к следующей задаче:

Задача. *Для изображения A известен его код T_A и некоторый симплексный набор индексов S. Требуется найти все правильные матрицы кода T_A относительно симплексного набора S.*

Заметим, что данную задачу можно решить очевидным способом: перебрать все матрицы кода $T^\mu \in M(T^\rho)$, проверяя для каждой из них условия правильной матрицы кода. Но такой способ оказывается неэффективным с вычислительной точки зрения, так как для числа N матриц кода верна оценка $2^{|A|} \leqslant N \leqslant 2^{(n+1)|A|}$, и потому прямой перебор приводит к экспоненциальной сложности по отношению к числу точек в изображении. Далее будет предложено более эффективное решение, имеющее линейную оценку сложности вычислений.

4. Вспомогательные леммы

Для квадратной матрицы A через $||A||$ обозначаем абсолютное значение ее определителя. Для произвольной матрицы A называем блоком $A_{R,C}$ подматрицу, образованную пересечением строк из множества R и столбцов из множества C. Пусть T_A — код изображения A; $S = (s_1, \ldots, s_{n+1})$ — некоторый симплексный набор индексов для кода T_A; R и C — два множества индексов таких, что $R \subseteq \mathbb{N}_{|A|}$, $C \subseteq \mathbb{N}_{n+1}$, $|R| = |C|$. Обозначим через $P_{S,R,C}$ такой набор индексов, который содержит все индексы из множества R и все индексы $s_j \in S$, для которых $j \notin C$. Из условия $|R| = |C|$ следует, что $P_{S,R,C} \in \mathbb{N}_{|A|}^{n+1}$,

и можно рассмотреть элемент кода $\rho_S^{P_{S,R,C}}$. Хотя набор $P_{S,R,C}$ определяется с точностью до перестановки индексов, число $\rho_S^{P_{S,R,C}}$ при этом не меняется, что обеспечивает корректность его определения.

Лемма 4. *Матрица кода T^μ является правильной тогда и только тогда, когда для каждой ее строки выполняется соотношение (1) и для каждого ее квадратного блока $T^\mu_{R,C}$ выполняется соотношение $||T^\mu_{R,C}|| = \rho_S^{P_{S,R,C}}$.*

Доказательство. Требуется доказать, что в определении правильной матрицы кода выполнение соотношения $|\mu_Q^P| = \rho_Q^P$ для всех наборов $P, Q \in \mathbb{N}_{|A|}^{n+1}$ равносильно выполнению соотношения $||T^\mu_{R,C}|| = \rho_S^{P_{S,R,C}}$ для всех квадратных блоков $T^\mu_{R,C}$.

Покажем сначала, что верно соотношение $||T^\mu_{R,C}|| = |\mu_S^{P_{S,R,C}}|$. Действительно, каждая строка, соответствующая индексу $s_j \in S$, $j \notin C$, имеет вид $(0, \ldots, 0, 1, 0, \ldots, 0)$, где единица стоит на j-й позиции. Поэтому при вычислении числа $\mu_S^{P_{S,R,C}}$ по формуле (2) остается только минор, соответствующий блоку $T^\mu_{R,C}$. Итак, остается доказать равносильность условий $|\mu_Q^P| = \rho_Q^P$ и $|\mu_S^{P_{S,R,C}}| = \rho_S^{P_{S,R,C}}$.

Если в наборе P существуют одинаковые индексы, то выполняется тождество $|\mu_Q^P| \equiv \rho_Q^P \equiv 0$. Если же набор P состоит из попарно различных индексов, то, очевидно, существуют множества R и C такие, что $P = P_{S,R,C}$. Отсюда следует, что

$$|\mu_S^{P_{S,R,C}}| = \rho_S^{P_{S,R,C}},\ (\forall R, C, |R| = |C|) \quad \Leftrightarrow \quad |\mu_S^P| = \rho_S^P,\ (\forall P \in \mathbb{N}_{|A|}^{n+1}).$$

Из соотношений $\mu_Q^P = \mu_S^P / \mu_S^Q$ и $\rho_Q^P = \rho_S^P / \rho_S^Q$ следует, что

$$|\mu_S^P| = \rho_S^P,\ (\forall P \in \mathbb{N}_{|A|}^{n+1}) \quad \Leftrightarrow \quad |\mu_Q^P| = \rho_Q^P,\ (\forall P, Q \in \mathbb{N}_{|A|}^{n+1}).$$

Объединяя результаты, получаем требуемое:

$$|\mu_Q^P| = \rho_Q^P,\ (\forall P, Q \in \mathbb{N}_{|A|}^{n+1}) \ \Leftrightarrow \ |\mu_S^{P_{S,R,C}}| = \rho_S^{P_{S,R,C}},\ (\forall R, C, |R| = |C|).$$

Лемма 4 доказана.

Определение 7. Квадратная матрица размера $m \times m$, $m \geqslant 2$ называется *цепной матрицей*, если в каждой строке и в каждом столбце матрицы имеется ровно два ненулевых элемента.

Определитель цепной матрицы называем цепным определителем. Также будем говорить о *цепных минорах*.

Лемма 5. *Пусть T — цепная матрица, и $A, B \in M(T)$. Соотношение $||A|| = ||B||$ выполняется тогда и только тогда, когда число элементов, на которых матрицы A и B различаются, четно.*

Доказательство. Считаем, что матрица A имеет следующий вид:

$$\begin{pmatrix} a_{11} & a_{12} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & a_{23} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{m-2m-2} & a_{m-2m-1} & 0 \\ 0 & \cdots & 0 & 0 & a_{m-1m-1} & a_{m-1m} \\ a_{m1} & 0 & \cdots & 0 & 0 & a_{mm} \end{pmatrix}$$

Цепную матрицу всегда можно привести к такому виду с помощью перестановки строк и столбцов, что не меняет абсолютного значения ее определителя. Нетрудно видеть, что для абсолютного значения определителя матрицы A верно:

$$||A|| = |a_{11} \cdot \ldots \cdot a_{mm} - a_{12} \cdot \ldots \cdot a_{m-1m} a_{m1}|.$$

Очевидно, изменение знаков у четного числа элементов a_{ij} не изменит значение $||A||$; наоборот, изменение знаков у нечетного числа элементов a_{ij} меняет значение на

$$|a_{11} \cdot \ldots \cdot a_{mm} + a_{12} \cdot \ldots \cdot a_{m-1m} a_{m1}|.$$

Лемма 5 доказана.

Лемма 6. *Пусть T — некоторая матрица. Рассмотрим матрицы $A = (a_{ij})$ и $B = (b_{ij})$, $A, B \in M(T)$. Если все соответствующие цепные миноры матриц A и B совпадают по абсолютному значению, то $A \sim B$. Обратно, если $A \sim B$, то все (не только цепные) соответствующие миноры матриц A и B совпадают по абсолютному значению.*

Доказательство. Второе утверждение следует из того, что при умножении строк или столбцов матрицы на -1 абсолютные значения ее миноров не меняются.

Докажем первое утверждение. Рассмотрим матрицу различия $D = (d_{ij})$, которая определяется следующим образом:

$$d_{ij} = \begin{cases} 0, & a_{ij} = b_{ij} = 0; \\ +1, & a_{ij} = b_{ij} \neq 0; \\ -1, & a_{ij} = -b_{ij} \neq 0. \end{cases}$$

Очевидно, $A \sim B$ тогда и только тогда, когда умножением на -1 некоторых строк и столбцов матрицы D можно получить матрицу без элементов -1. Заметим, что соответствующие цепные миноры матриц A и B совпадают по абсолютному значению тогда и только тогда, когда в соответствующей цепной подматрице матрицы D имеется четное число элементов -1 (это следует из леммы 5). Таким образом, достаточно доказать следующее утверждение: если все цепные подматрицы матрицы D содержат четное число элементов -1, то умножением на -1 некоторых строк и столбцов матрицы D можно получить матрицу без элементов -1.

Если матрица D состоит из одной строки, то утверждение верно. Пусть это утверждение верно для всех матриц с меньшим числом строк, чем у матрицы D. Докажем, что утверждение верно для матрицы D.

Пусть $\hat{D}$ — подматрица матрицы D, образованной всеми строками матрицы D начиная со второй строки. По предположению для подматрицы $\hat{D}$ утверждение верно. Умножая на -1 некоторые строки и столбцы матрицы D, исключим все элементы -1 в подматрице $\hat{D}$. Заметим, что при таких преобразованиях четность числа элементов -1 в цепных подматрицах матрицы D не изменится.

Обозначим через R множество всех строк, а через C множество всех столбцов матрицы D. Сформируем последовательность множеств строк $R_1, R_2, \ldots$ и последовательность множеств столбцов $C_1, C_2, \ldots$. Множество R_1 состоит только из первой строки матрицы D. Множество C_1 состоит из всех таких столбцов, что блок D_{R_1,C_1} состоит только из элементов -1. Здесь и далее мы предполагаем, что в первой строке существуют элементы -1, иначе матрица D уже приведена к требуемому виду. Множество C_{i+1} состоит из всех таких столбцов множества $C \setminus (\bigcup\limits_{j=1}^{i} C_j)$, что блок $D_{R_i,C_{i+1}}$ не содержит нулевых столбцов, $i \geqslant 1$. Множество R_{i+1} состоит из всех таких строк мно-

жества $R \setminus (\bigcup_{j=1}^{i} R_j)$, что блок $D_{R_{i+1},C_{i+1}}$ не содержит нулевых строк, $i \geqslant 1$.

Введем обозначения $\widetilde{R}_i = R \setminus (\bigcup_{j=1}^{i} R_j)$ и $\widetilde{C}_i = C \setminus (\bigcup_{j=1}^{i} C_j)$. Тогда верно:

Блоки $D_{\widetilde{R}_{i+1},C_{i+1}}$ и $D_{R_i,\widetilde{C}_{i+1}}$ состоят только из нулевых элементов, $i \geqslant 1$. (*)

Предположим, что при $i \geqslant 2$ существует блок D_{R_i,C_1}, содержащий хотя бы один ненулевой элемент. Тогда можно выбрать строки $r_1 \in R_1, \ldots, r_i \in R_i$ и столбцы $c_1 \in C_1, \ldots, c_i \in C_i$, так что блок $D_{\{r_1,\ldots,r_i\},\{c_1,\ldots,c_i\}}$ образует цепную подматрицу матрицы D, содержащую единственный элемент -1 (на пересечении строки r_1 и столбца c_1). Это приводит к противоречию с предположением о четности числа элементов -1. Таким образом, верно:

Блоки D_{R_i,C_1} состоят только из нулевых элементов, $i \geqslant 2$. (**)

Так как множества R и C конечны, то последовательности $R_1, R_2, \ldots$ и $C_1, C_2, \ldots$ также конечны. Пусть в первой последовательности k элементов: $R_1, \ldots, R_k$; во второй последовательности m элементов: $C_1, \ldots, C_m$. Из невозможности формирования множества R_{k+1} следует, что блок $D_{\widetilde{R}_k,C_{k+1}}$ состоит только из нулевых элементов, либо пуст. Аналогично, из невозможности формирования множества C_{m+1} следует, что блок $D_{R_m,\widetilde{C}_m}$ состоит только из нулевых элементов, либо пуст. Учитывая свойства (*) и (**), получаем:

Блоки $D_{(\bigcup_{j=2}^{k} R_j),C_1}$, $D_{\widetilde{R}_k,(\bigcup_{j=2}^{m} C_j)}$ и $D_{(\bigcup_{j=1}^{k} R_j),\widetilde{C}_m}$ состоят только из нулевых элементов, либо пусты. (***)

Умножим на -1 каждую строку из множества $\widetilde{R}_k$ и каждый столбец из множеств C_1 и $\widetilde{C}_m$. При этом преобразовании знаки элементов в блоках $D_{\widetilde{R}_k,C_1}$ и $D_{\widetilde{R}_k,\widetilde{C}_m}$ не изменятся, а знаки элементов в блоке D_{R_1,C_1} изменят знак. Все остальные элементы в рассматриваемых

строках и столбцах равны нулю (в случае непустоты соответствующих множеств) в силу свойства (***). Таким образом, после данных преобразований все элементы -1 будут исключены из матрицы D.

Итак, индукцией по числу строк матрицы D получаем требуемое утверждение. Лемма 6 доказана.

5. Алгоритм

Вернемся к задаче о построении всех правильных матриц кода на основе кода T_A и симплексного набора S. Далее описывается алгоритм, который строит некоторое множество матриц кода. В теореме 1 доказывается, что это множество состоит из правильных матриц кода и только из них. В теореме 2 доказывается, что алгоритм имеет линейную временную сложность по отношению к числу точек в изображении.

Используемые обозначения:
T_A — код изображения A.
S — симплексный набор индексов.
ρ_j^p — элементы ключевого подмножества кода T_A относительно симплексного набора S.
M^p — подмножество $\mathbb{N}_{n+1}$, определяемое как $M^p = \{j \in \mathbb{N}_{n+1} | \rho_j^p \neq 0\}$.
$G = (V, E)$ — вспомогательный граф. Каждая вершина $p \in V$ соответствует множеству M^p. Ребро $(p, q) \in E$ проводится, если $M^p \cap M^q \neq \varnothing$.
Объединение множеств M^p, соответствующих вершинам некоторой связной компоненты графа G, называем также связной компонентой и обозначаем через K.
$\mathfrak{K}$ — множество связных компонент K.
$\mathcal{P}$ — путь в графе G.
M, J — вспомогательные множества индексов из $\mathbb{N}_{n+1}$.
$T^\mu, T^{\hat{\mu}}$ — матрицы кода.
$\mu_j^p, \hat{\mu}_j^p$ — элементы матриц кода.
R — подмножество индексов из $\mathbb{N}_{|A|}$, соответствующих строкам матрицы кода.
C — подмножество индексов из $\mathbb{N}_{n+1}$, соответствующих столбцам матрицы кода.
$T^\mu_{R,C}$ — блок матрицы кода T^μ, соответствующий множествам R и C.
$\rho_S^{P_{S,R,C}}$ — элемент кода T_A, соответствующий множествам R и C.

$\sigma = \{\sigma_j = \pm 1 | j = 1, \ldots, n+1\}$ — произвольный набор из ± 1 длины $n+1$.
$\mathfrak{D}$ — множество матриц кода, полученных в результате работы алгоритма.

Алгоритм состоит из двух этапов и содержит несколько вспомогательных процедур. Содержание алгоритма представлено в виде псевдокода: после названия процедур указываются входные и выходные данные для этих процедур; оператор присваивания обозначается через «:=»; сравнение вещественных чисел (как и их представление) производится с некоторой точностью.

Алгоритм: $T_A, S \to \mathfrak{D}$
{
  Этап 1 — построение начальной матрицы кода: $T_A, S \to T^\mu$.
  Этап 2 — построение множества матриц кода: $T^\mu \to \mathfrak{D}$.
}

Этап 1: $T_A, S \to T^\mu$
{
  Инициализация: $V := \varnothing$, $E := \varnothing$, $\mathfrak{K} := \varnothing$.
  Цикл L_1 по индексам $p \in \mathbb{N}_{|A|}$:
    Процедура «Сформировать строку»: $T_A, S, p \to \{\mu_j^p\}_{j=1}^{n+1}, M^p$.
    Инициализируем вспомогательное множество: $M := \varnothing$.
    Цикл L_2 по компонентам $K \in \mathfrak{K}$:
      Если $M^p \cap K \neq \varnothing$, то:
        Обновляем вспомогательное множество: $M := M^p \cap K$.
        Выбираем произвольный индекс $j' \in M$.
        Инициализируем вспомогательное множество: $J := \{j'\}$.
        Цикл L_3 пока $J \neq M$:
          Выбираем произвольный индекс $j'' \in M \setminus J$.
          Процедура «Сформировать путь»: $G, j', j'' \to \mathcal{P}$.
          Процедура «Сформировать множества»: $\mathcal{P}, M, J, p, j' \to R, C, j$.
          Если $||T^\mu_{R,C}|| \neq \rho_S^{P_{S,R,C}}$, то: $\mu_j^p := -\rho_j^p$.
          Обновляем данные: $j' := j$, $J := J \cup \{j\}$.
        Конец цикла L_3.
      Конец если.
    Конец цикла L_2.
    Если $M \neq M^p$, то:

Процедура «Обновить компоненты»: $\mathfrak{K}, M^p \to \mathfrak{K}$.
Процедура «Обновить граф»: $G, M^p \to G$.
Конец если.
Конец цикла L_1.
}

Комментарии. В процедуре «Сформировать строку» элементы матрицы кода инициализируются элементами ключевого подмножества кода. В процедуре «Сформировать путь» выбирается цепочка множеств со специальными свойствами. На ее основе в процедуре «Сформировать множества» выбираются два множества индексов, определяющих квадратный блок в матрице кода. Далее вычисляется абсолютное значение определителя блока и сравнивается с соответствующим элементом кода. От результата сравнения зависит выбор знаков в элементах матрицы кода. Если множество M^p не содержится ни в одной компоненте связности целиком, то после цикла L_2 выполняется неравенство $M \neq M^p$. Верно и обратное. Таким образом, условие $M \neq M^p$ равносильно изменению множества компонент $\mathfrak{K}$ — в этом случае запускаются процедуры «Обновить компоненты» и «Обновить граф». В итоге будет получена некоторая матрица кода, которая поступает на вход этапа 2.

Этап 2: $T^\mu \to \mathfrak{D}$
{
Инициализация: $\mathfrak{D} := \varnothing$.
Цикл L_1 по наборам σ:
Цикл L_2 по индексам $p \in \mathbb{N}_{|A|}$:
Для каждого j полагаем $\hat{\mu}_j^p := \sigma_j \mu_j^p$, $\sum = \sum\limits_{j=1}^{n+1} \hat{\mu}_j^p$.
Если $\sum = -1$, то меняем знаки $\hat{\mu}_j^p$ для всех j.
Иначе: если $\sum \neq 1$, то переходим к следующему набору σ.
Конец цикла L_2.
Если $T^{\hat{\mu}} \notin \mathfrak{D}$, то $\mathfrak{D} := \mathfrak{D} \cup \{T^{\hat{\mu}}\}$.
Конец цикла L_1.
}

Комментарии. Очередная матрица кода $T^{\hat{\mu}}$ получается из начальной матрицы кода T^μ с помощью умножения столбцов на ± 1 в соответствии с набором σ, а также изменения знаков некоторых строк так,

чтобы сумма элементов во всех строках равнялось 1. Если это невозможно, то рассматривается следующий набор σ.

Процедура «Сформировать строку»: $T_A, S, p \to \{\mu_j^p\}_{j=1}^{n+1}, M^p$
{
  Инициализация: $M^p := \varnothing$.
  Цикл по $j := 1, \ldots, n+1$:
    Формируем p-ю строку матрицы кода T^μ: $\mu_j^p := \rho_j^p$.
    Формируем множество M^p: если $\rho_j^p \neq 0$, то $M^p := M^p \cup \{j\}$.
  Конец цикла.
}

Процедура «Сформировать путь»: $G, j', j'' \to \mathcal{P}$
{
  Находим вершину $M^{q'} \in V$ графа G, такую что $j' \in M^{q'}$.
  Находим вершину $M^{q''} \in V$ графа G, такую что $j'' \in M^{q''}$.
  В графе G находим кратчайший путь между вершинами $M^{q'}$ и $M^{q''}$:
    $\mathcal{P} = (M^{q_1}, M^{q_2}, \ldots, M^{q_{l-1}}, M^{q_l})$, $q_1 = q'$, $q_l = q''$.
  Если $j' \in M^{q_2}$, то удаляем M^{q_1} из $\mathcal{P}$.
}

Процедура «Сформировать множества»: $\mathcal{P}, M, J, p, j' \to R, C, j$
{
  Инициализируем множества: $R := \{p\}$, $C := \{j'\}$.
  Цикл по вершинам M^{q_i} пути $\mathcal{P}$:
    Если M^{q_i} — не первая вершина, то:
      Выбираем $j \in M^{q_i} \cap M^{q_{i-1}}$ и полагаем $C := C \cup \{j\}$.
    Конец если.
    Добавляем индекс вершины: $R := R \cup \{q_i\}$.
    Если $(M \setminus J) \cap M^{q_i} \neq \varnothing$, то:
      Выбираем $j \in (M \setminus J) \cap M^{q_i}$ и полагаем $C := C \cup \{j\}$.
      Выходим из цикла.
    Конец если.
  Конец цикла.
}

Процедура «Обновить компоненты»: $\mathfrak{K}, M^p \to \mathfrak{K}$
{
  Инициализация новой компоненты: $K' := M^p$.
  Цикл по компонентам $K \in \mathfrak{K}$:

Если $K \cap M^p \neq \varnothing$, то $K' := K' \cup K$, $\mathfrak{K} := \mathfrak{K} \setminus \{K\}$.
Конец цикла.
Добавляем сформированную компоненту: $\mathfrak{K} := \mathfrak{K} \cup \{K'\}$.
}

Процедура «Обновить граф»: $G, M^p \to G$
{
$V := V \cup \{M^p\}$.
Цикл по $M^q \in (V \setminus \{M^p\})$:
Если $M^q \subseteq M^p$, то удаляем вершину M^q и все инцидентные ей ребра.
Иначе: если $M^q \cap M^p \neq \varnothing$, то $E := E \cup \{(q, p)\}$.
Конец цикла.
}

Лемма 7. *Рассмотрим работу алгоритма на этапе 1. Выполняются следующие свойства:*

1) *Для каждой итерации цикла L_3 множества индексов R и C таковы, что блок $T^{\mu}_{R,C}$ является цепной матрицей.*
2) *Для очередной итерации цикла L_3 обозначим через $\{j_1, j_2\}$ пару индексов из C, для которых $\mu^p_{j_1} \neq 0$ и $\mu^p_{j_2} \neq 0$. После окончания работы цикла L_3 множество пар $\{j_1, j_2\}$ образует связное семейство множеств, объединение которых составляет множество $M = M^p \cap K$.*
3) *Для каждой итерации цикла L_1 верно: $|\mathfrak{K}| \leqslant n+1$, $|V| \leqslant n+1$.*

Доказательство.

1) Множества R и C формируются в результате работы процедур «Сформировать путь» и «Сформировать множества». По построению $\mathcal{P}$ — кратчайший путь между двумя вершинами графа, поэтому пересекаться могут только соседние вершины: $M^{q_i} \cap M^{q_{i+1}} \neq \varnothing$. При этом начальная и конечная вершины пути пересекаются с множеством M^p. В процедуре «Сформировать множества» выбирается первая после начальной вершина, которая также пересекается с множеством M^p. Так образуется замкнутая цепь из множества M^p и участка пути $\mathcal{P}$, в которой только соседние вершины имеют непустое пересечение. Из каждого такого пересечения выбирается по одному индексу — так

образуется множество C. Индексы самих вершин цепи образуют множество R. Рассмотрим произвольный индекс $q \in R$. По определению множества M^q верно: $\mu_j^q \neq 0$ только для индексов $j \in M^q$. Но из всех индексов $j \in C$ ровно два содержатся в M^q — это индексы из пересечений с соседними вершинами цепи. Таким образом, в каждой строке блока $T^{\mu}_{R,C}$ содержится ровно два ненулевых элемента. Рассмотрим произвольный индекс $j \in C$. Тогда включение $j \in M^q$ выполняется ровно для двух индексов q — это индексы соседних вершин цепи. Таким образом, в каждом столбце блока $T^{\mu}_{R,C}$ содержится ровно два ненулевых элемента. Это и означает, что блок будет цепной матрицей.

2) В обозначениях алгоритма индекс j_1 — это индекс j', а индекс j_2 — это индекс j. На следующей итерации цикла L_3 индекс j' полагается равным индексу j, то есть индекс j_1 из очередной пары — это индекс j_2 из предыдущей пары. Это и означает, что пары $\{j_1, j_2\}$ образуют связное семейство множеств.
Множество $J \subseteq M$ формируется из индексов j_1, j_2. На каждой итерации цикла L_3 множество J пополняется очередным индексом j (то есть j_2). Цикл L_3 заканчивает работу, когда $J = M$. Таким образом, после окончания работы цикла L_3 объединение пар $\{j_1, j_2\}$ образует множество $M = M^p \cap K$.

3) Для каждой связной компоненты $K \in \mathfrak{K}$ верно включение $K \subseteq \mathbb{N}_{n+1}$. Кроме того, любые две компоненты не пересекаются, и это свойство сохраняется на всех итерациях цикла L_1. Учитывая, что каждая компонента содержит хотя бы один индекс, получаем оценку для числа связных компонент: $|\mathfrak{K}| \leqslant n + 1$.
Граф G строится таким образом, что каждая его вершина не содержится ни в одной связной компоненте оставшегося множества вершин. Поэтому связные компоненты графа можно разделить на два типа: содержащие только вершины M^q, для которых $|M^q| \geqslant 2$; содержащие единственную вершину, состоящую из одного индекса. Для компоненты K обозначим через V_K число вершин, а через N_K число индексов в данной компоненте. Покажем, что $V_K < N_K$ для каждой компоненты K первого типа. Докажем это индукцией по V_K. При $V_K = 1$ утверждение верно. Пусть $V_K < N_K$ при $V_K = k$. Докажем неравенство при $V_K = k+1$. Фиксируем произвольную вершину M^q компоненты

K и рассмотрим оставшиеся вершины. Учитывая упомянутое свойство графа и неравенство $|M^q| \geqslant 2$ получаем две возможности:

а) Оставшиеся вершины распадаются на несколько связных компонент $K_1, \ldots, K_t$, $t \geqslant 2$. По предположению индукции верно $V_{K_i} < N_{K_i}$, $i = 1, \ldots, t$. Так как $V_K = V_{K_1} + \ldots + V_{K_t} + 1$ и $t \geqslant 2$, то $V_K \leqslant N_{K_1} + \ldots + N_{K_t} - t + 1 < N_K$.

б) Оставшиеся вершины образуют одну связную компоненту K'. Тогда M^q не содержится в ней полностью, и потому верно неравенство $N_{K'} < N_K$. Так как $V_K = V_{K'} + 1$ и $V_{K'} < N_{K'}$, то $V_K \leqslant N_{K'} < N_K$.

Итак, индукцией по V_K доказано, что $V_K < N_K$ для всех компонент первого типа. Для компонент второго типа выполняется равенство $V_K = N_K = 1$, а потому для всех вершин графа верно

$$|V| = \sum V_K \leqslant \sum N_K \leqslant |\mathbb{N}_{n+1}| = n + 1.$$

Лемма 7 доказана.

Теорема 1. *Если множество индексированных чисел T_A является кодом некоторого изображения A, то в результате работы алгоритма будут получены все правильные матрицы кода T_A относительно симплексного набора S и только они.*

Доказательство. Сначала покажем, что на этапе 1 будет получена матрица кода T^μ, удовлетворяющая соотношению $||T^\mu_{R,C}|| = \rho_S^{P_{S,R,C}}$ для любого квадратного блока $T^\mu_{R,C}$.

По условию множество индексированных чисел T_A является кодом некоторого изображения A. Пусть $T^{\hat{\mu}}$ — правильная матрица кода, соответствующая ключевому подмножеству μ-кода изображения A (относительно симплексного набора S). Обозначим через T^μ_k матрицу, образованную первыми k строками матрицы T^μ. Аналогично определяем матрицу $T^{\hat{\mu}}_k$.

Предположим, что для некоторого $k \geqslant 1$ выполняется $T^\mu_k \sim T^{\hat{\mu}}_k$. Рассмотрим $(k+1)$-ю строку матрицы T^μ_{k+1}. Эта строка определяется на $(k+1)$-й итерации цикла по индексам $p \in \mathbb{N}_{|A|}$. Согласно пункту 1 леммы 7 в цикле по компонентам $K \in \mathfrak{K}$ алгоритм находит множества индексов R и C, для которых матрица $T^\mu_{R,C}$ является

цепной и выполняется соотношение $||T^{\mu}_{R,C}|| = \rho_S^{P_{S,R,C}}$. Из правильности матрицы кода $T^{\hat{\mu}}$ согласно лемме 4 следует аналогичное соотношение $||T^{\hat{\mu}}_{R,C}|| = \rho_S^{P_{S,R,C}}$. Таким образом, имеет место равенство $||T^{\mu}_{R,C}|| = ||T^{\hat{\mu}}_{R,C}||$. Так как $T^{\mu}_{k} \sim T^{\hat{\mu}}_{k}$, то можно умножить некоторые строки и столбцы матрицы T^{μ}_{k+1} на -1, так чтобы первые k строк матриц T^{μ}_{k+1} и $T^{\hat{\mu}}_{k+1}$ совпадали. Данное преобразование не меняет абсолютные значения миноров матрицы T^{μ}_{k+1}, поэтому соотношение $||T^{\mu}_{R,C}|| = ||T^{\hat{\mu}}_{R,C}||$ остается верным.

После данного преобразования в блоках $T^{\mu}_{R,C}$ и $T^{\hat{\mu}}_{R,C}$ будут совпадать все строки, соответствующие индексам $R \setminus \{k+1\}$. Рассмотрим строку блоков $T^{\mu}_{R,C}$ и $T^{\hat{\mu}}_{R,C}$, соответствующую индексу $k+1$. В ней имеется ровно два ненулевых элемента: $\mu^{k+1}_{j_1}$ и $\mu^{k+1}_{j_2}$ для блока $T^{\mu}_{R,C}$, $\hat{\mu}^{k+1}_{j_1}$ и $\hat{\mu}^{k+1}_{j_2}$ для блока $T^{\hat{\mu}}_{R,C}$. Из соотношения $||T^{\mu}_{R,C}|| = ||T^{\hat{\mu}}_{R,C}||$ согласно лемме 5 следует, что выполняются либо равенства $\mu^{k+1}_{j_1} = \hat{\mu}^{k+1}_{j_1}, \mu^{k+1}_{j_2} = \hat{\mu}^{k+1}_{j_2}$, либо равенства $\mu^{k+1}_{j_1} = -\hat{\mu}^{k+1}_{j_1}, \mu^{k+1}_{j_2} = -\hat{\mu}^{k+1}_{j_2}$. Используя обозначения для блоков, получаем:

$$T^{\mu}_{\{k+1\},\{j_1,j_2\}} = \pm T^{\hat{\mu}}_{\{k+1\},\{j_1,j_2\}}.$$

Данное соотношение выполняется для всех множеств R, C. Согласно пункту 2 леммы 7 множество пар $\{j_1, j_2\}$ является связным семейством множеств, объединение которых составляет множество $M = M^{k+1} \cap K$, откуда следует соотношение:

$$T^{\mu}_{\{k+1\},M} = \pm T^{\hat{\mu}}_{\{k+1\},M}.$$

Пусть связная компонента K образована множествами $M^{p_1}, \ldots, M^{p_l}$. Если $T^{\mu}_{\{k+1\},M} = -T^{\hat{\mu}}_{\{k+1\},M}$, то умножим на -1 все столбцы матрицы T^{μ}_{k+1}, соответствующие компоненте K. После этого преобразования уже верно $T^{\mu}_{\{k+1\},M} = T^{\hat{\mu}}_{\{k+1\},M}$, но при этом $T^{\mu}_{\{p_1,\ldots,p_l\},K} = -T^{\hat{\mu}}_{\{p_1,\ldots,p_l\},K}$. Умножим на -1 все строки матрицы T^{μ}_{k+1}, соответствующие индексам $p_1, \ldots, p_l$. Теперь верно $T^{\mu}_{\{p_1,\ldots,p_l\},K} = T^{\hat{\mu}}_{\{p_1,\ldots,p_l\},K}$. Данные преобразования затрагивают только те ненулевые элементы матрицы T^{μ}_{k+1}, которые содержатся в блоках $T^{\mu}_{\{k+1\},M}$ и $T^{\mu}_{\{p_1,\ldots,p_l\},K}$.

Выполняя аналогичные преобразования для всех связных компонент $K \in \mathfrak{K}$, получим матрицу T^{μ}_{k+1}, которая может отличаться от

матрицы $T^{\hat{\mu}}_{k+1}$ только в тех столбцах j, для которых $j \notin K$ при всех $K \in \mathfrak{K}$. Но тогда $\mu^p_j = 0$ при $p \neq k+1$. Если $\mu^{k+1}_j = -\hat{\mu}^{k+1}_j$, то умножим на -1 столбец j матрицы T^{μ}_{k+1}. Так получим матрицу T^{μ}_{k+1}, совпадающую с матрицей $T^{\hat{\mu}}_{k+1}$.

Итак, матрица T^{μ}_{k+1} может быть преобразована в матрицу $T^{\hat{\mu}}_{k+1}$ путем умножения некоторых ее строк и столбцов на -1, что означает эквивалентность матриц T^{μ}_{k+1} и $T^{\hat{\mu}}_{k+1}$. Мы показали, что если для некоторого $k \geqslant 1$ выполняется $T^{\mu}_k \sim T^{\hat{\mu}}_k$, то выполняется и $T^{\mu}_{k+1} \sim T^{\hat{\mu}}_{k+1}$. Учитывая очевидную эквивалентность $T^{\mu}_1 \sim T^{\hat{\mu}}_1$, по индукции получаем $T^{\mu} \sim T^{\hat{\mu}}$. Но тогда для любых квадратных блоков $T^{\mu}_{R,C}$ и $T^{\hat{\mu}}_{R,C}$ верно $||T^{\mu}_{R,C}|| = ||T^{\hat{\mu}}_{R,C}||$. Согласно лемме 4 для матрицы $T^{\hat{\mu}}$ выполняется соотношение $||T^{\hat{\mu}}_{R,C}|| = \rho_S^{P_{S,R,C}}$, поэтому и для матрицы T^{μ} выполняется соотношение $||T^{\mu}_{R,C}|| = \rho_S^{P_{S,R,C}}$, что и требовалось доказать.

Далее покажем, что на этапе 2 будут получены все правильные матрицы кода и только они.

На этапе 1 алгоритм находит матрицу кода T^{μ}, удовлетворяющую соотношению $||T^{\mu}_{R,C}|| = \rho_S^{P_{S,R,C}}$ для любого квадратного блока $T^{\mu}_{R,C}$. Пусть для некоторой матрицы $T^{\hat{\mu}}$ также выполняется соотношение $||T^{\hat{\mu}}_{R,C}|| = \rho_S^{P_{S,R,C}}$ для любого квадратного блока $T^{\hat{\mu}}_{R,C}$. Это равносильно соотношению $||T^{\hat{\mu}}_{R,C}|| = ||T^{\mu}_{R,C}||$, которое в свою очередь равносильно условию $T^{\hat{\mu}}_{R,C} \sim T^{\mu}_{R,C}$ согласно лемме 6.

С учетом леммы 4 получаем, что все правильные матрицы кода — это матрицы $T^{\hat{\mu}}$, эквивалентные матрице T^{μ} и удовлетворяющие соотношению (1). Именно эти условия и проверяет алгоритм в цикле по σ. Теорема 1 доказана.

Следствие 1. *Существует не более 2^n различных классов a'-эквивалентных изображений с кодом T_A.*

Доказательство. Согласно утверждению 1 о биекции между множеством классов изображений и множеством правильных матриц кода достаточно доказать, что для некоторого (произвольного) симплексного набора S число правильных матриц кода T_A относительно набора S не превосходит числа 2^n. Для этого воспользуемся приведенным выше алгоритмом.

Все правильные матрицы кода и только они вычисляются на втором этапе алгоритма. Поэтому их число не превосходит числа итераций цикла по σ, а потому и числа наборов σ, равного 2^{n+1}. Но если для некоторого набора σ получена правильная матрица кода $T^{\hat{\mu}}$, то и для набора $-\sigma$ будет получена та же правильная матрица кода $T^{\hat{\mu}}$ (изменяя знаки строк, получаем тот же результат). Таким образом, оценка улучшается в два раза и число правильных матриц кода не превосходит числа 2^n. Следствие 1 доказано.

Теорема 2. *Существует реализация алгоритма с временной сложностью $C(n)|A|$, где $C(n) = O(n4^n + n^4)$.*

Доказательство. Рассмотрим следующее представление данных для используемых в алгоритме объектов [3]. Код, матрицы кода и симплексный набор индексов представляются с помощью массивов. Множества M^p, K, M, J представляются с помощью битовых массивов длины $n+1$, что позволяет формировать объединение, пересечение и разность множеств за время $O(n)$, и определять принадлежность элемента множеству за время $O(1)$. Множество $\mathfrak{K}$ компонент K представляется с помощью связного списка, что позволяет выполнять доступ или удаление произвольной компоненты K за время $O(n)$, а добавление новой компоненты за время $O(1)$. Аналогично с помощью связных списков представляются путь $\mathcal{P}$, множества индексов R и C, множество $\mathfrak{D}$. Граф $G = (V, E)$ представляется с помощью $|V|$ списков смежностей, что позволяет добавлять новую вершину за время $O(1)$, добавлять новое ребро или удалять вершину с инцидентными ей ребрами за время $O(|V|)$.

Оценим время выполнения каждой процедуры:

Процедура «Сформировать строку». Каждая итерация цикла выполняется за время $O(1)$, всего итераций $n+1$. В итоге получаем время $O(n)$.

Процедура «Сформировать путь». Согласно пункту 3 леммы 7 имеем $|V| \leqslant n+1$. Таким образом, за время $O(|V|) = O(n)$ будут просмотрены все вершины графа и выбраны те две вершины $M^{q'}, M^{q''}$, которые содержат индексы j', j'' соответственно. Кратчайший путь находим с помощью поиска в ширину [4], который выполняется за время $O(|V|^2) = O(n^2)$. Проверка и последующее удаление первой вершины пути производится за время $O(|V|) = O(n)$. В итоге получаем время $O(n^2)$.

Процедура «Сформировать множества». Инициализация множеств R, C происходит за время $O(1)$. В теле цикла основные вычисления относятся к операциям над битовыми массивами, что приводит к оценке $O(n)$. Число итераций цикла не больше, чем вершин в найденном пути, и оценивается числом $n+1$. В итоге получаем время $O(n^2)$.

Процедура «Обновить компоненты». Основные вычисления производятся в цикле. Тело цикла выполняется за время $O(n)$. Согласно пункту 3 леммы 7 имеем $|\mathfrak{K}| \leqslant n+1$, что дает соответствующую оценку для числа итераций цикла. В итоге получаем время $O(n^2)$.

Процедура «Обновить граф». Тело цикла выполняется за время $O(n)$, количество итераций цикла не больше $n+1$. В итоге получаем время $O(n^2)$.

Оценим время выполнения этапа 1:

Цикл L_3. Процедуры «Сформировать путь» и «Сформировать множества» выполняются за время $O(n^2)$. Вычисление определителя блока $T^{\mu}_{R,C}$ можно осуществить за время $O(|R|^3) = O(n^3)$, используя метод Гаусса [5]. Остальные операции выполняются за время $O(n)$. Число итераций цикла не превосходит мощности множества M. В итоге получаем время $O(n^3)|M|$.

Цикл L_2. За счет цикла L_3 время выполнения тела цикла L_2 оценивается как $O(n^3)|M|$. Суммируя эту величину по всем компонентам $K \in \mathfrak{K}$ и учитывая, что $\cup_K M = \cup_K (K \cap M^p) = M^p$, получаем время $O(n^3)|M^p| = O(n^4)$.

Цикл L_1. Процедура «Сформировать строку» выполняется за время $O(n)$. Цикл L_2 выполняется за время $O(n^4)$. Процедуры «Обновить компоненты» и «Обновить граф» выполняются за время $O(n^2)$. Остальные операции выполняются за время $O(n)$. Число итераций цикла L_1 равно $|A|$. В итоге получаем время $O(n^4)|A|$, которое и дает оценку для времени выполнения этапа 1.

Оценим время выполнения этапа 2:

Тело цикла L_2 выполняется за время $O(n)$. Число итераций цикла L_2 не превосходит числа $|A|$. Число итераций цикла L_1 равно 2^{n+1}. Проверка совпадения двух матриц кода выполняется за время $(n+1)|A|$. На i-й итерации цикла L_1 нужно провести не более i таких проверок на совпадение, что дает в сумме по всем итерациям порядка $(2^{n+1})^2$ проверок. В итоге для этапа 2 получаем время $O(n4^n)|A|$.