2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

doi 10.26089/NumMet.v22r419 УДК 519.17:519.6

# Математическая модель и алгоритм вычисления циклов ячеек карты графа

**Б. Н. Иванов**
*Дальневосточный федеральный университет (ДВФУ),*
*Институт математики и компьютерных технологий,*
*Владивосток, Российская Федерация*
*ORCID: 0000-0002-4904-9759, e-mail: ibn8826@mail.ru*

**Аннотация:** Выделенные свойства циклов *DFS-базиса* блока карты простого графа позволили составить математическую модель вычисления циклов ячеек карты графа. По данной модели предложен практический алгоритм вычисления циклов ячеек карты простого графа. Алгоритм имеет квадратичную сложность относительно числа вершин в графе.

**Ключевые слова:** карта графа, ячейки карты, циклы графа, свойства циклов.

**Для цитирования:** Иванов Б.Н. Математическая модель и алгоритм вычисления циклов ячеек карты графа // Вычислительные методы и программирование. 2021. **22**, № 4. 294–305. doi 10.26089/NumMet.v22r419.

# Mathematical model and algorithm for calculating the cycles of the cells of the graph map

**Boris N. Ivanov**
*Far Eastern Federal University,*
*Institute of Mathematics and Computer Technologies, Vladivostok, Russia*
*ORCID: https://orcid.org/0000-0002-4904-9759, e-mail: ibn8826@mail.ru*

**Abstract:** The selected properties of the cycles of the *DFS-basis* block of a simple graph map allowed us to create a mathematical model for calculating the cycles of the cells of the graph map. According to this model, a practical algorithm for calculating the cycles of the graph map cells is proposed. The algorithm has a quadratic complexity relative to the number of vertices in the graph.

**Keywords:** graph map, map cells, graph cycles, cycle properties.

**For citation:** B. N. Ivanov, "Mathematical model and algorithm for calculating the cycles of the cells of the graph map," Numerical Methods and Programming. **22** (4), 294–305 (2021). doi 10.26089/NumMet.v22r419.

**1. Введение.** Рассматриваемая математическая модель вычисления циклов ячеек карты графа реализована как решение задачи раскраски различного рода территорий на поверхности Земли, в частности государств, в рамках действующей геоинформационной системы "Океан" [1]. Исходные данные в задаче раскраски территорий — это география поверхности Земли (береговая черта) и границы государств. Линии границ разбивают береговую черту на отрезки, которые дополняют границы государств. Данной модели поверхности Земли поставим в соответствие конечный *простой планарный* граф $G = (X, U)$, где $U$ — ребра графа (границы государств), $X$ — вершины графа (точки пересечения границ). К такому графу всегда можно свести рассматриваемую модель поверхности Земли, включая фиктивные вершины на границах или береговой черте.

2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

Изучение циклической структуры графа является важной составляющей общего анализа структуры графа. Первый практический алгоритм перечисления всех циклов графа был опубликован в работе *Welch* [2]. Модифицированная версия алгоритма [2] была опубликована в статье *Gibbs* [3]. Работы [2, 3] определили направление разработки алгоритмов генерации циклов графов. Было показано, что в данной задаче метод поиска с возвращениями позволяет в среднем существенно понизить сложность полного перебора и перейти к практически значимым алгоритмам.

После этого был предложен ряд других алгоритмов [4, 5] порождения *всех циклов графа*. Обзор этих алгоритмов и их относительных достоинств можно найти в работе [6]. Основу этих алгоритмов составляет *метод поиска в глубину* на графе [7]. Это адаптированный к структуре графа поиск с возвращениями. Сложность генерации циклов в таких алгоритмах определяется величиной $O((|X| + |U|)(\ell + 1))$, где $\ell$ — число порожденных циклов.

Перечисление циклов графа с выделенными свойствами, как правило, осуществляется на основе алгоритмов генерации всего множества циклов. Каждый такой алгоритм является уникальным. Проверка каких-либо условий может существенно увеличить сложность алгоритма.

Сегодня можно встретить и параллельные алгоритмы генерации циклов в графе [8]. В большей степени такие алгоритмы представляют интерес в теоретическом плане. Авторы пытаются решать тестовые задачи большой размерности, используя параллельные вычисления. В этом случае для поиска циклов в работе [9] предлагается приближенный алгоритм как альтернатива полному перебору. Для информационных сетей актуальна задача сетевой надежности передачи данных, и здесь первостепенное значение приобретает знание циклической структуры сети [10].

Генерация циклов в простом планарном графе рассматривалась автором данной статьи в работе [11], в которой дается геометрическое обоснование вычислительных формул с ограничениями по виду графа. В настоящей статье основную часть материала составляет вывод и обоснование расчетных формул циклов ячеек карты простого планарного графа. Алгоритм вычисления по расчетным формулам дается с привязкой к блоковой структуре графа. Для данного алгоритма даются оценки его верхней границы сложности вычислений, которая является квадратичной относительно числа вершин в графе.

**2. Постановка задачи.** Будем пользоваться терминами и обозначениями, принятыми в теории графов. *Простой граф* — это неориентированный граф без петель, любая пара вершин которого соединена не более чем одним ребром. В таком графе ребра представляются неупорядоченными парами вершин [12, с. 12]. Маршрут по графу является *циклом*, если он замкнут и каждое его ребро встречается в нем не более одного раза, вершины в цикле могут повторяться. *Цикл является простым*, если в нем никакая вершина не повторяется [12, с. 35].

Укладка на плоскости ребер *простого планарного* графа без самопересечений определяет разбиение такой плоскости, называемое *планарным подразбиением* или *картой* графа [13, с. 31]. Карта графа фиксирует на плоскости геометрию ребер графа. Сама карта складывается из простых многоугольников (подразбиений), которые будем называть *ячейками* карты. Каждая сторона такой ячейки может принадлежать еще ровно одной такой же ячейке. Границы ячеек карты являются простыми циклами.

В данной статье решается задача вычисления циклов ячеек карты графа $G = (X, U)$ по данным множеств его вершин $X$ и ребер $U$. Новизна предлагаемого решения состоит в том, что циклы ячеек карты представляются в виде линейных комбинаций циклов *DFS-базиса* (DFS — от Depth First Search) графа. Искомые линейные комбинации циклов строятся явно на основе выделенных свойств циклов базиса. Необходимые свойства циклов базиса для составления линейных комбинаций последовательно доказываются в следующих разделах статьи. В отдельный раздел вынесено определение линейного векторного пространства циклов и его базиса. В разделе 6 рассматривается алгоритм вычисления циклов ячеек карты графа с найденными линейными комбинациями циклов базиса. Сложность алгоритма вычисления всех циклов ячеек карты графа является квадратичной относительно числа вершин в графе.

**3. Фундаментальное множество циклов карты графа.** Пусть $G = (X, U)$ — конечный связный простой планарный граф. Обозначим через $M = \{G_1, G_2, \ldots, G_{n_M}\}$ множество всех его подграфов $G_i = (X, U_i)$, где $U_i \subseteq U$. На множестве $M$ определим бинарную операцию $\oplus$ так, что $\forall G_i, G_j \in M : G_i \oplus G_j = (X, U_i \oplus U_j)$, где символ $\oplus$ обозначает операцию симметрической разности, $U_i \oplus U_j = \{u \,|\, u \in U_i \cup U_j \wedge u \notin U_i \cap U_j\}$. Множество $M$ с операцией $\oplus$ является абелевой группой. Единица группы — пустой подграф $O = (X, \varnothing)$. Обратным к $G_k$ является тот же самый $G_k$, так как $G_k \oplus G_k = O$.

2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

Обозначим через $Z = \{Z_1, Z_2, \ldots, Z_{n_Z}\}$, где $Z_k = (X, z_k)$ — подграфы графа $G$, $z_k$ — все простые циклы графа $G$. Пусть $L = \{\lambda_{i_1} Z_{k_1} \oplus \lambda_{i_2} Z_{k_2} \oplus \ldots \oplus \lambda_{i_Z} Z_{k_Z}\}$ — линейная оболочка циклов множества $Z$, где $\lambda_{i_r} \in \{0, 1\}$, $0 \cdot Z_{k_r} = O$ и $1 \cdot Z_{k_r} = Z_{k_r}$. Множество $L$ является линейным векторным пространством *циклов* над полем $P =< \{0, 1\}, \oplus, \cdot >$ [14, с. 61]. Уточним структуру, размерность и базис пространства $L$.

Пусть $T_0 = (X, U_0)$ — произвольное остовное дерево исходного графа $G = (X, U)$. Для дерева выполняется соотношение $|U_0| = |X| - 1$. Пусть $E = U \setminus U_0 = \{e_1, e_2, \ldots, e_{n_F}\}$ — ребра, не вошедшие в $T_0$. Такие ребра называются обратными. Любое обратное ребро $e_i \in E$ порождает в $T_0$ ровно один *простой цикл* $C_i$, где $e_i \in C_i$ и $e_i \notin C_j$, если $i \neq j$. Таких циклов $C_i$ можно составить $n_F = |U \setminus U_0| = |U| - |X| + 1$. Множество $F = \{C_1, C_2, \ldots, C_{n_F}\}$ называется *фундаментальным множеством циклов* и составляет базис пространства $L$ [14, с. 63]. В простом планарном графе [14, с. 36] выполняется соотношение $|U| \leqslant 3|X| - 6$, что дает основание пользоваться оценкой для числа циклов $n_F = O(|X|)$.

**4. Свойства циклов *DFS-базиса* карты графа.** Построение фундаментального множества циклов выполним *методом поиска в глубину* со стеком. Подробное изложение данного алгоритма встречается во многих работах. Оригинальное его описание представлено в статье [15]. Адаптированный вариант алгоритма для практического применения рассматривается в работе [16, с. 385]. При порождении фундаментального множества циклов методом поиска в глубину на простом связном графе $G = (X, U)$ строится дерево поиска, которое является остовным деревом графа и называется *DFS-деревом*. Каждое обратное ребро порождает один цикл относительно этого дерева. Все множество таких циклов составляет *DFS-базис* циклов графа. При поиске остовного дерева используется поиск в глубину *со стеком*, в котором хранятся все текущие вершины пройденного пути в данный момент. Когда попадаем на обратное ребро, обнаруженный цикл будет состоять из этого ребра и ребер, соединяющих вершины из верха стека до вершины обратного ребра в стеке.

**Определение 1 (смежных циклов).** Простые циклы $H_1$, $H_2$ будем называть *смежными*, если их пересечение $H_1 \cap H_2$ состоит из одного непустого непрерывного участка ребер. Тогда $H_1 \oplus H_2$ есть простой цикл.

**Утверждение 1.** *Взаимное расположение каждой пары циклов базиса $C_1$ и $C_2$ таково, что если $C_1 \cap C_2 \neq \varnothing$, то либо $C_1$ и $C_2$ — смежные, либо $C_1$ и $C_2$ имеют ровно одну общую вершину.*

**Доказательство.** Допустим, что пересечение $C_1 \cap C_2 \neq \varnothing$ содержит две или более вершины. Пусть $e_1 \in C_1$ и $e_2 \in C_2$ — обратные ребра данных циклов, $e_1 \notin C_2$ и $e_2 \notin C_1$. Пройдя по циклу $C_1$ в обе стороны от ребра $e_1$, мы найдем две вершины $x_1$ и $x_2$, принадлежащие также $C_2$ (рис. 1). Обозначим через $C_1(x_1, e_1, x_2)$, $C_2(x_1, e_2, x_2)$ участки циклов $C_1$ и $C_2$, содержащие соответственно ребра $e_1$ и $e_2$. Их объединение $C_1(x_1, e_1, x_2) \cup C_2(x_1, e_2, x_2)$ есть простой цикл.

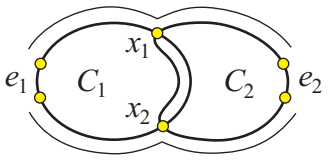
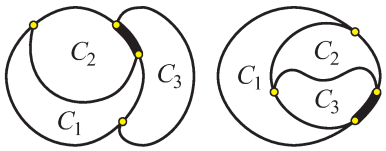



Рис. 1. Взаимное расположение циклов базиса

Fig. 1. Mutual arrangement of basis cycles

Обозначим через $R_1(x_1, x_2)$ и $R_2(x_1, x_2)$ участки циклов соответственно $C_1$ и $C_2$, не вошедших в простой цикл $C_1(x_1, e_1, x_2) \cup C_2(x_1, e_2, x_2)$. Если допустить, что $R_1(x_1, x_2) \neq R_2(x_1, x_2)$ (рис. 1), то вершины $x_1$ и $x_2$ будут связаны четырьмя маршрутами. Тогда при удалении всех обратных ребер в графе получим остовное дерево $T_0$, в котором вершины $x_1$ и $x_2$ останутся связанными двумя маршрутами $R_1(x_1, x_2) \neq R_2(x_1, x_2)$ (рис. 1), а значит, в $T_0$ будет цикл. Это противоречит тому, что $T_0$ есть дерево. Итак, имеем $R_1(x_1, x_2) = R_2(x_1, x_2)$, следовательно, $C_1$ и $C_2$ есть смежные циклы, а $C_1 \oplus C_2$ — простой цикл, составленный из участков $C_1(x_1, e_1, x_2)$ и $C_2(x_1, e_2, x_2)$.




*а)* *б)*

Рис. 2. Укладки трех циклов при наличии у них общей границы

Fig. 2. Laying of three cycles if they have a common border

**Определение 2 (нулевой вложенности).** Пусть $C_v$ и $C_{v_j}$ — циклы базиса карты графа. Будем говорить, что цикл $C_{v_j}$ имеет нулевую вложенность в цикле $C_v$, если цикл $C_{v_j}$ вложен в $C_v$ и не вложен ни в какой другой цикл из $C_v$.

**Утверждение 2.** *Пусть циклы базиса $C_1$, $C_2$, $C_3$ имеют общую границу ребер. Из утверждения 1 имеем, что такие циклы являются попарно смежными. Геометрия компоновки на карте указанных циклов допускает только два случая. В первом случае цикл $C_2$ имеет нулевую вложенность относительно цикла $C_1$, а цикл $C_3$ не вложен в $C_1$ (рис. 2 а). Во втором случае цикл $C_3$ имеет нулевую*

2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

*вложенность в цикле $C_2$, а цикл $C_2$ имеет нулевую вложенность в цикле $C_1$ (рис. 2 б). В любом случае 1) среди циклов $C_1$, $C_2$, $C_3$ существует цикл ($C_2$) с нулевой вложенностью относительно другого цикла ($C_1$); 2) среди циклов $C_1$, $C_2$, $C_3$ нет двух циклов с нулевой вложенностью относительно третьего цикла.*

**Определение 3 (скелета графа).** Пусть $C_v$ и $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ — циклы базиса карты графа, где $C_{v_i}$ — циклы нулевой вложенности в цикле $C_v$. Для данной совокупности циклов определим неориентированный граф $Q_v = (X_v, U_v)$, вершинами которого $v_i \in X_v$ являются циклы $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$; ребро $(v_i, v_j) \in U_v$, если циклы $C_{v_i}, C_{v_j}$ смежные. Если граф $Q_v = (X_v, U_v)$ связный, то циклы $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ будем называть *смежными в совокупности*. Граф смежных в совокупности циклов $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ будем обозначать через $J_v = J_v(C_{v_1}, C_{v_2}, \ldots, C_{v_k})$, а граф $Q_v = (X_v, U_v)$ будем называть его *скелетом*.

**Утверждение 3.** *Если циклы DFS-базиса $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ карты графа нулевой вложенности в цикле $C_v$ определяют связный граф смежности $J_v(C_{v_1}, C_{v_2}, \ldots, C_{v_k})$, то скелет данного графа $Q_v$ имеет структуру дерева.*

**Доказательство.** Так как граф смежности $J_v$ является связным графом, то и его скелет $Q_v$ по построению будет связным графом. Покажем, что скелет $Q_v$ является деревом. Допустим, что в скелете $Q_v$ существует цикл $(C_{r_1}, C_{r_2}, \ldots, C_{r_m}, C_{r_1})$, для цикла должно выполняться условие $m \geqslant 3$. Тогда в данной циклической последовательности у каждого цикла $C_{r_j}$ всегда найдутся два смежных ему цикла $C_{r_{j-1}}$ и $C_{r_{j+1}}$. Смежные циклы *DFS-базиса* при их формировании размещаются в стеке циклов последовательно (см. алгоритм поиска в глубину в работе [16, с. 385]). Пусть при формировании данные циклы в стеке имели следующий порядок: $C_{i_1}, C_{i_2}, \ldots, C_{i_m}$. Цикл $C_{i_m}$ размещается в стеке последним, все остальные циклы уже разместили свои начала ребер в стеке. По предположению у цикла $C_{i_m}$ есть два смежных ему цикла, например, $C_{i_k}$ и $C_{i_l}$, начала которых уже находятся в стеке. Отсюда заключаем, что начало ребер цикла $C_{i_m}$ будет общим участком для циклов $C_{i_k}$ и $C_{i_l}$. Итак, три цикла $C_{i_m}$, $C_{i_k}$ и $C_{i_l}$ будут иметь общий участок ребер. Из утверждения 2 заключаем, что один из данных циклов является вложенным в другой. Это противоречит условию, что исходные циклы имеют нулевую вложенность в цикле $C_v$. Значит, скелет $Q_v$ графа $J_v$ имеет структуру дерева.

**Утверждение 4.** *Удаление обратных ребер в графе $J_v$ смежных в совокупности циклов базиса $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ карты графа сводит граф $J_v$ к дереву.*

**Доказательство.** Покажем, что после удаления обратных ребер граф $J_v$ останется связным и без циклов. Удаление обратных ребер разрушит только одну сторону в каждом из циклов $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ и сохранит переход по другой стороне цикла от одного его смежного участка до другого по ребрам циклов (обратные ребра не могут находиться в смежной части циклов базиса). Следовательно, после удаления обратных ребер граф смежности $J_v$ останется связным. С другой стороны, удаление обратных ребер в исходном графе сводит его к остовному дереву $T_0$. Следовательно, и в графе $J_v$ не может быть циклов после удаления в нем обратных ребер.

**Утверждение 5.** *Пусть циклы DFS-базиса $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ карты графа нулевой вложенности в цикле $C_v$ определяют связный граф смежности $J_v(C_{v_1}, C_{v_2}, \ldots, C_{v_k})$. Если граф смежности $J_v$ имеет две общие вершины $x_i$ и $x_j$ с циклом $C_v$, то данные вершины могут принадлежать только одному циклу графа смежности.*

**Доказательство.** Допустим, что в графе смежности $J_v$ вершины $x_i$ и $x_j$ принадлежат разным циклам $x_i \in C_i$, $x_j \in C_j$ и $x_i \notin C_j$, $x_j \notin C_i$. Удаление обратных ребер в исходном графе сведет его к остовному дереву $T_0$, а граф $J_v$ сведет к дереву $T_0(C_v)$ (см. утверждение 4). Следовательно, после удаления обратных ребер вершины $x_i$ и $x_j$ останутся связанными двумя маршрутами: один маршрут $M_1$ по неразрушенной стороне цикла $C_v$, а другой маршрут $M_2$ по дереву $T_0(C_v)$. Если указанные маршруты не совпадают ($M_1 \neq M_2$), то в дереве $T_0$ будет цикл $M_1 \cup M_2$, что противоречит дереву $T_0$. Другой случай — маршруты совпадают ($M_1 = M_2$). Маршрут $M_2$ по дереву $T_0(C_v)$ связывает последовательно попарно смежные циклы $C_{i_1}, C_{i_2}, \ldots, C_{i_r}$ ($2 \leqslant r \leqslant k$), а так как $M_1 = M_2$, то каждый из этих циклов является смежным циклу $C_v$. Тогда будем иметь циклическую последовательность из трех попарно смежных циклов $(C_v, C_{i_1}, C_{i_2}, C_v)$. Такие циклы *DFS-базиса* должны иметь общий участок ребер (см. доказательство в общем случае в утверждении 3). Из утверждения 2 имеем, что среди таких трех циклов не существует двух циклов с нулевой вложенностью относительно третьего цикла. В нашем случае $C_{i_1}$ и $C_{i_2}$ являются циклами нулевой вложенности в $C_v$. И в этом случае получили противоречие. Значит, в графе смежности $J_v$ может существовать единственный цикл, который может иметь две общие вершины с циклом $C_v$.

2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

**Утверждение 6.** *Пусть циклы DFS-базиса $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ карты графа нулевой вложенности в цикле $C_v$ определяют связный граф смежности $J_v(C_{v_1}, C_{v_2}, \ldots, C_{v_k})$. Результатом слияния циклов данного графа будет один простой цикл $L_v = C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_k}$.*

**Доказательство.** Из утверждения 3 имеем, что скелетом данного графа смежности $J_v$ является дерево $Q_v$. Слияние простых циклов $C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_k}$ выполним в порядке их следования по структуре дерева $Q_v$ при обходе его сверху вниз. Положим в начале обхода цикл слияния $L_v$ равным $C_{v_k}$ — начало обхода дерева $Q_v$. Каждый раз, приходя в вершину $C_{v_j}$ по дереву $Q_v$, будем выполнять слияние циклов: $L_v = L_v \oplus C_{v_j}$. Пусть вершина $C_{v_j} \in X_v$ по дереву скелета $Q_v$ смежна ранее пройденной вершине $C_{v_i} \in X_v$. Слияние смежных простых циклов $C_{v_i} \oplus C_{v_j}$ дает один простой цикл. Так как скелет $Q_v$ является деревом, то из пройденных по нему вершин в текущий момент вершина $C_{v_i}$ является единственной вершиной, смежной вершине $C_{v_j}$. Поэтому результатом слияния $L_v = L_v \oplus C_{v_j}$ будет один простой цикл.

**Определение 4.** Два графа смежности $J_u$ и $J_w$ в цикле $C_v$ будем называть *несмежными*, если каждая пара циклов $C_{u_i} \in J_u$ и $C_{w_j} \in J_w$ является несмежной. Если же существует пара смежных циклов $C_{u_i}$ и $C_{w_j}$ таких, что $C_{u_i} \in J_u$ и $C_{w_j} \in J_w$, то по определению графов смежности будем иметь один граф смежности $J_u = J_w$.

**5. Блоковые свойства циклов *DFS-базиса* карты графа.** Предлагаемый алгоритм вычисления циклов ячеек карты графа ориентирован на блоковые свойства карты. Для выделения блоков карты графа можно воспользоваться алгоритмом, представленным в работе [7], модифицированным *методом поиска в глубину*. Далее отметим несколько блоковых свойств циклов *DFS-базиса* карты графа, которые составят основу алгоритма выделения циклов ячеек блока карты графа.

**Определение 5 (блока).** В блоке графа все ребра сильно циклически связаны [12, с. 109]. Это означает, что для любой пары ребер блока $u_1, u_2$ существует такая последовательность простых циклов $H_1, H_2, \ldots, H_k$, что $u_1 \in H_1$, $u_2 \in H_k$ и любая пара соседних циклов $H_i$, $H_{i+1}$ имеет по крайней мере одно общее ребро.

**Замечание 1.** Все простые циклы графа распределяются между блоками графа. Каждый простой цикл расположен целиком в своем блоке [12, с. 111]. Поэтому далее полагаем, что исходный граф $G = (X, U)$ есть невырожденный конечный блок карты графа (вырожденный блок состоит из одного ребра).

**Утверждение 7.** *Пусть карта графа $G = (X, U)$ является блоком, а $H$ — его простой цикл. Тогда подграф $B(H) = (X_H, U_H)$, ограниченный циклом $H$, включая ребра цикла $H$, и подграф $B(G \setminus H)$ являются блоками, где под операцией разности $G \setminus H$ понимается удаление из $G$ ребер, вложенных в цикл $H$, а ребра цикла $H$ не удаляются.*

**Доказательство.** Докажем, что подграф $B(H) = (X_H, U_H)$ является блоком. Пусть $u, v \in U_H$ — произвольные два ребра. Тогда в графе $G$ существует простой цикл $P$, для которого $u, v \in P$ [12, с. 110]. Если цикл $P$ выходит за границы области цикла $H$, то существует и простой цикл в графе $B(H)$, содержащий ребра $u$ и $v$. Действительно, исходный граф — это карта графа, и все пересечения ребер цикла $P$ с ребрами цикла $H$ являются вершинами из $X_H$. В этом случае любую выступающую часть дуги цикла $P$ за область цикла $H$ заменим на часть дуги цикла $H$, которая замыкает выступающую часть дуги $P$. Новый цикл $P$ будет простым и локализованным в границах цикла $H$. Значит $B(H)$ есть блок [12, с. 110]. Доказательство того, что $B(G \setminus H)$ является блоком, повторяет рассмотренное доказательство для блока $B(H) = (X_H, U_H)$.

**Утверждение 8.** *Пусть циклы DFS-базиса $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ карты графа нулевой вложенности в цикле $C_v$ представляются совокупностью подграфов смежности $J_{t_1}, J_{t_2}, \ldots, J_{t_k}$, попарно не смежных друг с другом. Тогда каждый подграф $J_{t_j}$ является отдельным блоком.*

**Доказательство.** Каждый подграф $J_{t_j}$ является блоком, так как он составляется из циклов, смежных в совокупности (см. определение 5). Покажем, что подграфы $J_{t_i}$ и $J_{t_j}$ являются различными блоками. Если предположить, что $J_{t_i}$ и $J_{t_j}$ составляют один блок, то они должны иметь по крайней мере две общие вершины $x_1$ и $x_2$ [12, с. 111].

Рассмотрим остовное дерево $T_0$ исходного графа после удаления из него обратных ребер. В этом случае из утверждения 4 имеем, что каждый из подграфов $J_{t_i}$ и $J_{t_j}$ останется деревом. Значит, вершины $x_1$ и $x_2$ останутся связанными в подграфах $J_{t_i}$ и $J_{t_j}$. Пусть это будут маршруты $R_1(x_1, x_2)$ и $R_2(x_1, x_2)$. Если $R_1(x_1, x_2) \neq R_2(x_1, x_2)$, то они составят цикл. Это противоречит тому, что в дереве $T_0$ нет циклов. Следовательно, $R_1(x_1, x_2) = R_2(x_1, x_2)$, а значит, существует общее ребро $u$, принадлежащее подграфам $J_{t_i}$ и $J_{t_j}$. Тогда существует пара смежных циклов $u \in C_{v_i} \in J_{t_i}$ и $u \in C_{v_j} \in J_{t_j}$, что противоречит

2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

условию несмежности подграфов $J_{t_i}$ и $J_{t_j}$ (см. определение 4). Следовательно, предположение неверное и все подграфы $J_{t_j}$ являются отдельными блоками.

**Утверждение 9.** *Пусть $C_v$ — произвольный цикл DFS-базиса блока карты графа $G$, $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ — весь перечень нулевой вложенности в цикле $C_v$, которые разбиваются на совокупность попарно не смежных графов смежности $J_{t_1}, J_{t_2}, \ldots, J_{t_k}$. Тогда выполняются следующие условия:*

*1) в каждом графе $J_{t_j}$ существует единственный цикл базиса $C_{t_j} \in J_{t_j}$, смежный циклу $C_v$;*

*2) циклы $C_v$ и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ являются смежными в совокупности;*

*3) общими вершинами графов $J_{t_j}$ могут быть только вершины цикла $C_v$.*

**Доказательство.** Исходный граф $G$ является блоком карты графа. Из утверждения 7 имеем, что 1) подграф $B(C_v)$, ограниченный циклом $C_v$, есть блок; 2) подграф $B(C_v \setminus C_{v_1})$, ограниченный циклами $C_v$ и $C_{v_1}$, есть блок; 3) подграф $B((C_v \setminus C_{v_1}) \setminus C_{v_2})$, ограниченный циклами $C_v$ и $C_{v_1}, C_{v_2}$, есть блок; и т.д., 4) подграф $B((((C_v \setminus C_{v_1}) \setminus C_{v_2}) \setminus \ldots) \setminus C_{v_n})$, ограниченный циклами $C_v$ и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$, есть блок. Из определения операции разности (см. утверждение 7) заключаем, что последний блок $B((((C_v \setminus C_{v_1}) \setminus C_{v_2}) \setminus \ldots) \setminus C_{v_n})$ состоит только из ребер цикла $C_v$ и ребер циклов $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$. По условию данного утверждения множество всех циклов $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ распределяется между графами смежности $J_{t_1}, J_{t_2}, \ldots, J_{t_k}$, каждый из которых не смежный с другим.

Таким образом, блок $B((((C_v \setminus C_{v_1}) \setminus C_{v_2}) \setminus \ldots) \setminus C_{v_n})$ состоит из блока $C_v$ и различных блоков $J_{t_1}, J_{t_2}, \ldots, J_{t_k}$ (см. утверждение 8). Объединение составляющих блоков в один блок возможно, если каждый из блоков $J_{t_j}$ имеет с циклом $C_v$ по крайней мере две общие вершины [12, с. 111].

Пусть $J_{t_j}$ и $C_v$ имеют две общие вершины $x_1$ и $x_2$. Из утверждения 5 имеем, что данные вершины $x_1$ и $x_2$ могут принадлежать лишь одному циклу $C_{t_j}$ графа смежности $J_{t_j}$, а из утверждения 1 следует, что такие циклы $C_{t_j}$ и $C_v$ являются смежными (*условие 1 доказано*).

Отсюда заключаем, что цикл $C_v$ объединяет циклы графов смежности $J_{t_1}, J_{t_2}, \ldots, J_{t_k}$ в одну совокупность смежных циклов, т.е. циклы $C_v$ и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ являются смежными в совокупности (*условие 2 доказано*).

*Доказательство условия 3.* Имеем, что графы $J_{t_i}$ и $J_{t_j}$ смежные с циклом $C_v$. Предположим, что графы $J_{t_i}$ и $J_{t_j}$ имеют хотя бы одну общую вершину $x_0$, отличную от вершин цикла $C_v$. Удаление обратных ребер в блоке $B(C_v)$ не нарушит связности графов $J_{t_i}$ и $J_{t_j}$ и будет существовать маршрут от $y_1$ по ребрам $J_{t_i}$ через $x_0$ по ребрам $J_{t_j}$ в $y_2$, где $y_1 \neq y_2$ — вершины смежных участков графа $J_{t_i}$ с циклом $C_v$ и графа $J_{t_j}$ с циклом $C_v$. Удаление обратного ребра в цикле $C_v$ не нарушит и его связности — останется переход по нему от $y_1$ до $y_2$, а в остовном дереве $T_0(C_v)$ будет цикл. Следовательно, предположение неверное, и графы $J_{t_i}$ и $J_{t_j}$ не могут иметь общих вершин, кроме вершин на цикле $C_v$.

**Утверждение 10.** *Пусть $C_v$ — произвольный цикл DFS-базиса блока карты графа и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ — весь перечень нулевой вложенности в него циклов базиса. Тогда цикл*

$$S_v = C_v \oplus C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_n} \tag{1}$$

*определяет одну ячейку карты графа, локализованной в цикле $C_v$.*

**Доказательство.** Из утверждения 9 имеем, что циклы $C_v$ и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ являются смежными в совокупности. Результатом слияния данных циклов по формуле (1) будет один цикл $S_v = C_v \oplus C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_n}$ (см. утверждение 6). Структурно цикл $S_v$ есть цикл $C_v$, из которого удалили все вложенные в него циклы $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$. Цикл $S_v$ не может быть пустым, в этом случае будем иметь линейно зависимые циклы $C_v$ и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$, что противоречит их принадлежности *DFS-базису* циклов. Предположение, что остаток $S_v$ может оказаться объединением двух ячеек карты или более, нарушит условие полноты циклов *DFS-базиса*. Значит, $S_v$ есть цикл ячейки блока карты графа. Все множество циклов ячеек блока карты графа $S = \{S_1, S_2, \ldots, S_{n_F}\}$ составляет базис циклов блока. Каждый цикл *DFS-базиса* $C_j$ блока можно представить как линейную комбинацию $C_j = S_{j_1} \oplus S_{j_2} \oplus \ldots \oplus S_{j_k}$ циклов ячеек блока, которые охватывает данный цикл $C_j$. Размерность линейного пространства является *инвариантом*, следовательно, число циклов *DFS-базиса* и циклов ячеек карты совпадает. Отсюда заключаем, что все циклы ячеек карты $S_v$ представляются по формуле (1), где каждый цикл $C_v$ определяет свой цикл $S_v$.

**6. Вычисление циклов ячеек карты графа.** Предлагаемый алгоритм вычисления циклов ячеек карты графа $G = (X, U)$ реализует их расчет по формуле (1): $S_v = C_v \oplus C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_n}$, где $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ — перечень циклов нулевой вложенности в цикле $C_v$. Справедливость формулы (1) огра-

2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

ничивается блоками графа. Как отмечалось выше (см. раздел 5), при разложении исходного графа на блоки можно воспользоваться алгоритмом, представленным в работе [7].

Для вычислений по формуле (1) в разделе 6.1 рассматривается организация данных доступа к циклам базиса с учетом их уровней вложенности. Слияние циклов нулевой вложенности по формуле (1) вынесено в раздел 6.2. Перебор всех циклов базиса $C_v$ и вложенных в них циклов нулевой вложенности $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ вынесено в раздел 6.3.

**6.1. Структура вложенности циклов DFS-базиса блока карты графа.** Введем на множестве циклов *DFS-базиса* блока карты графа $F = \{C_1, C_2, \ldots, C_{n_F}\}$ бинарное отношение ($\prec$) *нулевой вложенности*, которое позволит практически реализовать доступ к перечню циклов $C_1, C_2, \ldots, C_{n_F}$ в соответствии с их уровнем вложенности.

**Определение 6.** Отношение $C_i \prec C_j$ выполняется, если $C_i \subset C_j$ и $\forall\, C_k \neq C_j \ : (C_i \subset C_k \Rightarrow C_j \subset C_k)$. Это означает, что $C_j$ — наименьший охватывающий цикл для $C_i$, тогда цикл $C_i$ будет иметь нулевую вложенность в цикле $C_j$.

Структуру вложенности циклов базиса $F = \{C_1, C_2, \ldots, C_{n_F}\}$ представим в виде *ориентированного графа циклов* $G_F = (X_F, U_F)$, где $X_F = \{C_1, C_2, \ldots, C_{n_F}\}$ — множество вершин, $U_F$ — множество ребер, ребра представляются парами вершин. Так, ориентированное ребро $u = (C_i, C_j) \in U_F$, если для циклов $C_i$ и $C_j$ выполняется отношение нулевой вложенности $C_i \prec C_j$. Направленное ребро $(C_i, C_j)$ ориентировано от вложенного цикла $C_i$ к наименьшему охватывающему его циклу $C_j$ ($C_i \to C_j$). В качестве наименьшего охватывающего цикла $C_j$ для цикла $C_i$ назначается тот, для которого выполняется условие

$$\min\{\, \mu(C_k) \,|\, C_i \subset C_k,\, k = 1, 2, \ldots, n_F\}, \tag{2}$$

где $\mu(C_k)$ — площадь, охватываемая циклом $C_k$. Для каждого цикла $C_i$ в векторе счетчиков $r_1, r_2, \ldots, r_{n_F}$ будем фиксировать его уровень вложенности, где $r_i$ — количество циклов базиса $C_k$, которые охватывают данный цикл $C_i$. Эффективные практические процедуры проверки вложенности циклов $C_i \prec C_j$ и вычисления значений $r_i$ уровней вложенности циклов $C_i$ можно найти в работе [17]. Сложность формирования вектора $r_1, r_2, \ldots, r_{n_F}$ по формуле (2) составляет $O(n_F^2) = O(|X|^2)$, где необходимо каждый цикл сравнить с каждым.

Введенное отношение нулевой вложенности ($\prec$) позволяет выполнить представление *корневых деревьев циклов* графа $G_F = (X_F, U_F)$ ориентированным реберным списком $Q_1, Q_2, \ldots, Q_{n_F}$, где элемент $Q_i = j$, если выполняется отношение $C_i \prec C_j$. Для циклов $C_i$, которые не имеют охватывающих циклов, значение $Q_i = 0$. Такими вершинами являются корни деревьев графа $G_F = (X_F, U_F)$. Корни деревьев определяются по вектору счетчиков $r_1, r_2, \ldots, r_{n_F}$. Для корней значения счетчиков $r_i = 0$. *Корневые деревья циклов* позволяют выполнить перебор циклов базиса $C_i$ по уровням их вложенности как сверху вниз, так и обратно.

В описании алгоритмов для представления неориентированного графа циклов $G_F = (X_F, U_F)$ будет использоваться его структура смежности $Adj[x]$ — множество вершин, смежных с данной вершиной $x \in X_F$, которое определяется для каждой вершины. Процедура преобразования реберного списка $Q_1, Q_2, \ldots, Q_{n_F}$ в структуру смежности $Adj[x]$ представлена в алгоритме 1.

Сложность формирования структуры смежности $Adj[x]$ по алгоритму 1 составляет $O(n_F^2) = O(|X|^2)$ как результат выполнения двух вложенных циклов.

Алгоритм 1. Преобразование реберного списка в структуру смежности

Algorithm 1. Converting an edge list to structure adjacencies

```
1: Procedure CreateAdj(Q, Adj)
2:   for x ∈ X_F do begin
3:     Adj[x] = ∅;    {Initial empty list for vertex x}
4:     if Q_x ≠ 0 then Adj[x] = {Q_x};    {The edge outgoing from the vertex x}
5:     for v ∈ X_F do begin
6:       if Q_v = x then Adj[x] = Adj[x] ∪ {v};    {Edges entering the vertex x}
7:     end;
8:   end;
9: end.
```

2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

**Замечание 2.** Корневая структура деревьев графа $G_F$ должна быть сохранена и в структуре смежности $Adj[x]$. Это будет выполнено, если в перечне $X_F$ корни деревьев будут встречаться раньше других вершин деревьев. Для этого необходимо упорядочить векторы структуры смежности $Adj[x]$, где $x \in X_F$, по возрастанию счетчиков вложенности циклов $r_1, r_2, \ldots, r_{_F}$. Если $r_x \leqslant r_y$, то в структуре смежности данные $Adj[x]$ для вершины $x$ должны располагаться раньше данных $Adj[y]$ для вершины $y$. Сложность сортировки данных относительно вектора $r_1, r_2, \ldots, r_{_F}$ можно считать равной $O(n_{_F}^2) = O(|X|^2)$.

В заключение суммируем, что сложность формирования структуры смежности $Adj[x]$ для графа циклов $G_F = (X_F, U_F)$ составляет $O(|X|^2)$.

**6.2. Слияние смежных в совокупности циклов базиса.** Вынесем из общего алгоритма 3 формирование циклов ячеек карты графа реализацию процедуры слияния циклов. Сама процедура слияния $Join(W_v, S_v)$ представлена в алгоритме 2. Параметрами процедуры выступают множество циклов $W_v = \{C_v, C_{v_1}, C_{v_2}, \ldots, C_{v_n}\}$ и результирующий цикл слияния $S_v$, где $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ — перечень циклов нулевой вложенности в цикле $C_v$. Начальное значение цикла слияния полагаем равным пустому.

Чтобы результатом слияния $S_v = S_v \oplus C_w$, где $C_w \in W_v$, на каждом шаге был ровно один цикл, необходимо, чтобы цикл $C_w$ был смежным с текущим циклом слияния $S_v$. Согласно утверждению 6 это возможно, если перебор циклов $C_w \in W_v$ выполнить одним из обходов скелета $Q_w = (X_w, U_w)$ графа смежности $J_w$ для совокупности циклов $W_v = \{C_v, C_{v_1}, C_{v_2}, \ldots, C_{v_n}\}$ (см. определение 3). Скелет $Q_w = (X_w, U_w)$ данного графа смежности $J_w$ является деревом (см. утверждения 3 и 9). Воспользуемся обходом скелета $Q_v$ сверху вниз. Процедура обхода сверху вниз заимствована из работы [7], в основе которой лежит обход графа в глубину.

В начале процедуры формируется структура смежности $Adj[x]$ представления дерева скелета $Q_w = (X_w, U_w)$ графа смежности $J_w$. Рекурсивная процедура $Depth(x)$ выполняет обход сверху вниз поддерева с корнем в вершине $x \in X_w$. Результатом обхода скелета $Q_w = (X_w, U_w)$ будет цикл слияния $S_v$ — цикл ячейки карты графа.

Сложность алгоритма 2 — это сложность выполнения процедуры $Join(W_v, S_v)$ и процедуры $Depth(x)$. Оценим сложность $Join(W_v, S_v)$ без учета процедуры $Depth(x)$. Ее сложность определяется двумя вложенными циклами, размерность каждого из которых равна $n_v = |W_v| \geqslant 1$. Следовательно, сложность процедуры составляет $O(n_v^2)$. Сама процедура $Join(W_v, S_v)$ выполняется для каждого цикла базиса $C_v$,

Алгоритм 2. Слияние множества $W_v$ смежных в совокупности циклов

Algorithm 2. Merging a set of $W_v$ adjacent in a collection of cycles

```
1:  Procedure Join (W_v, S_v)
2:     X_w = ∅;
3:     for i = 1 to |W_v| do X_w = X_w ∪ {i};    {Cycle numbers of the set W_v}
4:     for C_x ∈ W_v do begin    {Formation of Adj[x] skeleton tree Q_w = (X_w, U_w)}
5:        Adj[x] = ∅;    {Initial empty list Adj[x]}
6:        for C_y ∈ W_v do begin
7:           if (x ≠ y) ∧ (C_x ∩ C_y ≠ ∅) then begin
8:              Adj[x] = Adj[x] ∪ {y};    {Expand the list Adj[x]}
9:           end;
10:       end;
11:    end;
12:    z ∈ X_w;    {Start traversing the skeleton tree Q_w = (X_w, U_w)}
13:    S_v = C_z;    {Initial merge cycle }
14:    Depth(z);    {Traversing a skeleton subtree rooted at z ∈ X_w}
15: end.
16:
17: Procedure Depth (x)    {Top-down traversal of a skeleton subtree with a root in x ∈ X_w}
18:    for y ∈ Adj[x] do begin    {Enumeration of adjacent vertices in the skeleton}
19:       S_v = S_v ⊕ C_y;    {Merging of adjacent cycles}
20:       Depth(y);    {Traversing a skeleton subtree rooted at y ∈ X_w}
21:    end;
22: end.
```

2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

число которых равно $n_F$. Таким образом, суммарная сложность выполнения процедуры $Join(W_v, S_v)$ в алгоритме 3 составит $O\left(\sum\limits_{v=1}^{n_F} n_v^2\right)$. Заметим, что каждый цикл $C_v$ один раз входит в $W_v$ и может входить еще один раз в другое множество $W_u$, определяемое циклом $C_u$, для которого цикл $C_v$ является циклом нулевой вложенности. Отсюда имеем, что сумма $\sum\limits_{v=1}^{n_F} |W_v| = \sum\limits_{v=1}^{n_F} n_v = m \leqslant 2n_F$.

**Утверждение 11.** *Сумма $\sum\limits_{i=1}^{k} n_i^2$ при условии, что $\sum\limits_{i=1}^{k} n_i = m$ и $n_i \geqslant 1$, принимает максимальное значение на том же наборе значений переменных $n_1, n_2, \ldots, n_k$, что и сумма $\sum\limits_{i=1}^{k} C_{n_i}^2$, где $C_{n_i}^2$ — число сочетаний из $n_i$ по 2, $C_{n_i}^2 = n_i(n_i-1)/2$ (полагаем $C_{n_i}^2 = 0$ при $n_i = 1$).*

**Доказательство.** Ясно, что суммы $\sum\limits_{i=1}^{k} n_i^2$ и $\left(\sum\limits_{i=1}^{k} n_i^2 - m\right)/2$ принимают максимальные значения на одних и тех же наборах $n_1, n_2, \ldots, n_k$. С другой стороны, сумма $\left(\sum\limits_{i=1}^{k} n_i^2 - m\right)/2 = \left(\sum\limits_{i=1}^{k} n_i^2 - \sum\limits_{i=1}^{k} n_i\right)/2 = \sum\limits_{i=1}^{k} (n_i^2 - n_i)/2 = \sum\limits_{i=1}^{k} n_i(n_i-1)/2 = \sum\limits_{i=1}^{k} C_{n_i}^2$.

**Утверждение 12.** *Пусть простой граф $G$ имеет $m$ вершин и $k$ компонент связности $G_i$, $n_i$ — число вершин в $G_i$, $\sum\limits_{i=1}^{k} n_i = m$. Максимальное число ребер в таком графе $G$ равно $\sum\limits_{i=1}^{k} C_{n_i}^2$, если каждая из компонент $G_i$ является полным графом. Из работы [12, с. 38] имеем, что максимальное число ребер в графе $G$ с $m$ вершинами и $k$ компонентами связности равно $C_{m-k+1}^2$, если граф состоит из $k-1$ изолированной вершины и одного полного подграфа с $m-k+1$ вершиной. В этом случае набор значений $(n_1, n_2, \ldots, n_k) = (m-k+1, 1, \ldots, 1)$.*

Вернемся к оценке максимального значения суммы $\sum\limits_{v=1}^{n_F} n_v^2$. Из утверждений 11 и 12 имеем, что максимальное значение суммы $\sum\limits_{v=1}^{n_F} n_v^2$ достигается на наборе $(n_1, n_2, \ldots, n_F) = (m - n_F + 1, 1, \ldots, 1)$. Ее значение на этом наборе равно $(m-n_F+1)^2+1^2+\ldots+1^2 = (m-n_F+1)^2+n_F-1 \leqslant (2n_F-n_F+1)^2+n_F-1 = (n_F+1)^2+n_F-1 = O(n_F^2)$. Таким образом, $\sum\limits_{v=1}^{n_F} n_v^2 = O\left(n_F^2\right) = O(|X|^2)$.

Процедура $Depth(x)$ имеет линейную сложность $O(n_v)$ — это сложность обхода дерева. Сама процедура $Depth(x)$ в алгоритме 3 будет также выполняться $n_F$ раз. Тогда ее суммарная сложность составит $\sum\limits_{v=1}^{n_F} O(n_v) = O(n_F) = O(|X|)$, так как $\sum\limits_{v=1}^{n_F} n_v \leqslant 2n_F$. Значит, суммарная сложность выполнения процедуры $Join(W_v, S_v)$ в алгоритме 3 составит $O(|X|^2)$.

**6.3. Вычисление циклов ячеек блока карты графа.** Исходный граф циклов $G_F = (X_F, U_F)$ задается структурой смежности $Adj[x]$, формирование ее рассматривалось в алгоритме 1.

Расчет циклов $S_v$ ячеек карты графа выполняется по формуле (1): $S_v = C_v \oplus C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_n}$. Последовательность вычисления циклов $S_v$ имеет существенное значение. В рассматриваемой задаче перебор циклов $C_v$ наиболее оптимально выполнить обходом снизу вверх *корневых деревьев* циклов базиса (см. раздел 6.1). Суть данного обхода составляет процедура, заимствованная из работы [7]. В этом случае обработке цикла $C_v$ будет предшествовать перебор вложенных в него циклов $C_{v_j}$. Данный подход реализован в алгоритме 3 формирования циклов ячеек карты графа.

Чтобы отличить уже пройденные вершины, вводится вектор меток вершин $Mark[v]$, значения которых равно 1 для пройденных вершин и 0 — для не пройденных. Цикл $C_v$ и его циклы нулевой вложенности $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ формируют множество $W$ смежных в совокупности циклов. Слияние циклов множества $W$ процедурой $Join(W, S_v)$ дает цикл $S_v$ ячейки карты графа. Все выделенные циклы $S_v$ составят искомое множество $S = \{S_1, S_2, \ldots, S_{n_F}\}$ циклов ячеек блока карты графа.

Сложность алгоритма 3 — это сложность обхода дерева, которая является линейной относительно числа вершин $n_F = |X|$. В алгоритме для каждой вершины $x \in X_F$ выполняется процедура слияния $Join(W, S_x)$. Суммарная сложность ее выполнения в этом алгоритме вычислена выше в разделе 6.2 и составляет $O\left(|X|^2\right)$. Следовательно, сложность предложенного алгоритма вычисления циклов ячеек карты графа является квадратичной $O\left(|X|^2\right)$ относительно числа вершин исходного графа.

2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

Алгоритм 3. Формирование множества $S$ циклов ячеек блока карты графа

Algorithm 3. Formation of the set $S$ of cycles of the cells of the graph map block

```
1:  S = ∅;    {Initial empty set of cells of the graph map block}
2:  for x ∈ X_F do Mark[x] = 0;    {Not traversed vertices labels}
3:  for x ∈ X_F do if Mark[x] = 0 then begin    {Finding the start of a crawl}
4:  Root(x);    {Traversing a rooted tree rooted at x}
5:  end;
6:
7:  Procedure Root (v)    {Traversing the tree from bottom to top with a root at the top v}
8:     Mark[v] = 1;    {Vertice passed}
9:     W = ∅;    {Initial empty set of nested cycles in C_v}
10:    for x ∈ Adj[v] do begin    {Enumeration of adjacent vertices}
11:       if Mark[x] = 0 then begin    {Search for a failed vertice}
12:          Root(x);    {Traversing a subtree rooted at x}
13:          W = W ∪ {C_x};    {Add nested cycle C_x}
14:       end;
15:    end;
16:    W = W ∪ {C_v};    {Add a cycle. C_v — tree root}
17:    Join(W, S_v);    {S_v — the result of merging the cycles of the set W}
18:    S = S ∪ {S_v};    {Save cycle of cell S_v in set S}
19: end.
```

**6.4. Формирование общего списка циклов ячеек карты графа.** В рамках предложенной модели вычисление всех циклов ячеек карты графа выполняется по блокам графа. Блоковая структура графа может иметь вложенный характер. Например, один блок вложен в цикл другого блока. Заполнение цветами такого рода циклов в задаче раскраски карты на поверхности земли может приводить к их потере на карте, если заполнить вложенный цикл раньше цикла, который его охватывает.

Пусть $S_1, S_2, \ldots, S_n$ — вычисленные циклы ячеек графа по всем блокам. Сформируем линейную структуру вложенности циклов $S_1, S_2, \ldots, S_n$, которая удовлетворяет свойству, что вложенные циклы по этому списку следования всегда располагаются правее циклов, охватывающих их. Для каждого цикла $S_i$ в векторе счетчиков $r_1, r_2, \ldots, r_n$ будем фиксировать его уровень вложенности, где $r_i$ — количество циклов $S_k$, которые охватывают данный цикл $S_i$ (см. раздел 6.1).

Для размещения последовательности циклов $S_1, S_2, \ldots, S_n$ в порядке следования, отвечающего структуре их вложенности, достаточно выполнить их сортировку согласно возрастанию значений счетчиков $r_1, r_2, \ldots, r_n$. Теперь последовательное заполнение циклов $S_1, S_2, \ldots, S_n$ будет отвечать свойству, что каждый вложенный цикл будет размещаться поверх охватывающего цикла. Формирование значений $r_1, r_2, \ldots, r_n$ подробно рассматривается в разделе 6.1. Сложность данной процедуры — это сложность сортировки. Сложность сортировки данных относительно вектора $r_1, r_2, \ldots, r_n$ можно принять равной $O(n^2)$. Для карты графа $O(n^2) = O(|X|^2)$, где $X$ — множество вершин графа.

**7. Заключение.** Предложена математическая модель вычисления циклов ячеек карты графа. Выделены и доказаны все необходимые свойства циклов *DFS-базиса* данной модели. Предложен практический алгоритм реализации математической модели вычисления циклов ячеек блока карты графа. Сложность каждого шага данного алгоритма не превышает квадратичной сложности $O(|X|^2)$ относительно числа вершин в графе.

### Список литературы


1. *Иванов Б.Н.* Решение задачи расчета оптимальных маршрутов судов в рамках геоинформационной системы “ОКЕАН” // Вычислительные методы и программирование. 2012. **13**, № 1. 226–234.
2. *Welch J.T.* A mechanical analysis of the cyclic structure of undirected linear graphs // J. Assoc. Comput. Mach. 1966. **13**, N 2. 205–210.
3. *Gibbs N.E.* A cycle generation algorithm for finite undirected linear graphs // J. Assoc. Comput. Mach. 1969. **16**, N 4. 564–568.
4. *Tarjan R.* Enumeration of the elementary circuits of a directed graph // SIAM J. Comput. 1973. **2**, N 3. 211–216.

2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

5. *Jonson D.B.* Finding all the elementary circuits of a directed graph // SIAM J. Comput. 1975. **4**, N 1. 77–84.
6. *Mateti P., Deo N.* On algorithms for enumerating all circuits of a graph // SIAM J. Comput. 1976. **5**, N 1. 90–99.
7. *Tarjan R.* Depth-first search linear graph algorithms // SIAM J. Comput. 1972. **1**, N 2. 146–160.
8. *Mahdi F., Safar M., Mahdi K.* Detecting cycles in graphs using parallel capabilities of GPU // Digital Information and Communication Technology and Its Applications. Vol. 167. Berlin: Springer, 2011. 193–205.
9. *Kavitha T., Mehlhorn K., Michail D.* New approximation algorithms for minimum cycle bases of graphs // Algorithmica. 2011. **59**, N 4. 471–488.
10. *Pfaltz J.L.* Chordless cycles in networks // IEEE 29th International Conference on Data Engineering Workshops. New York: IEEE Press, 2013. 223–228.
11. *Иванов Б.Н.* Генерация циклов ячеек карты простого планарного графа // Вычислительные методы и программирование. 2014. **15**, № 2. 304–316.
12. *Оре О.* Теория графов. М.: Наука, 1980.
13. *Препарата Ф., Шеймос М.* Вычислительная геометрия. Введение. М.: Мир, 1989.
14. *Алексеев В.Е., Таланов В.А.* Графы. Модели вычислений. Структуры. Нижний Новгород: Изд-во ННГУ, 2005.
15. *Paton K.* An algorithm for finding a fundamental set of cycles of a graph // Commun. ACM. 1969. **12**, N 9. 514–518.
16. *Рейнголд Э., Нивергельд Ю., Део Н.* Комбинаторные алгоритмы. Теория и практика. М.: Мир, 1980.
17. *Иванов Б.Н.* Структуры вложенности поля изолиний в задаче градиентного заполнения // Вычислительные методы и программирование. 2006. **7**, № 2. 30–40.

*Поступила в редакцию*
*13 сентября 2021*

*Принята к публикации*
*5 ноября 2021 г.*

### Информация об авторе

*Борис Николаевич Иванов* — к.ф.-м.н., доцент, Дальневосточный федеральный университет (ДВФУ), Институт математики и компьютерных технологий, ул. Суханова, 8, 690950, Владивосток, Российская Федерация.

### References

1. B. N. Ivanov, “Solution of the Optimal Ship Route Problem in the Framework of the OKEAN Geoinformation System,” Vychisl. Metody Programm. **13** (1), 226–234 (2012).
2. J. T. Welch, “A Mechanical Analysis of the Cyclic Structure of Undirected Linear Graphs,” J. Assoc. Comput. Mach. **13** (2), 205–210 (1966).
3. N. E. Gibbs, “A Cycle Generation Algorithm for Finite Undirected Linear Graphs,” J. Assoc. Comput. Mach. **16** (4), 564–568 (1969).
4. R. Tarjan, “Enumeration of the Elementary Circuits of a Directed Graph,” SIAM J. Comput. **2** (3), 211–216 (1973).
5. D. B. Jonson, “Finding all the Elementary Circuits of a Directed Graph,” SIAM J. Comput. **4** (1), 77–84 (1975).
6. P. Mateti and N. Deo, “On Algorithms for Enumerating all Circuits of a Graph,” SIAM J. Comput. **5** (1), 90–99 (1976).
7. R. Tarjan, “Depth-First Search Linear Graph Algorithms,” SIAM J. Comput. **1** (2), 146–160 (1972).
8. F. Mahdi, M. Safar, and K. Mahdi, “Detecting Cycles in Graphs Using Parallel Capabilities of GPU,” in *Digital Information and Communication Technology and Its Applications* (Springer, Berlin, 2011), Vol. 167, pp. 193–205.
9. T. Kavitha, K. Mehlhorn, and D. Michail, “New Approximation Algorithms for Minimum Cycle Bases of Graphs,” Algorithmica **59** (4), 471–488 (2011).
10. J. L. Pfaltz, “Chordless Cycles in Networks,” in *Proc. IEEE 29th Int. Conf. on Data Engineering Workshops, Brisbane, Australia, April 8–12, 2013* (IEEE Press, New York, 2013), pp. 223–228.
11. B. N. Ivanov, “Generation of Cycles of Map Cells for a Simple Planar Graph,” Vychisl. Metody Programm. **15** (2), 304–316 (2014).
12. O. Ore, *Theory of Graphs* (AMS Press, Providence, 1962; Nauka, Moscow, 1980).
13. F. P. Preparatat and M. I. Shamos, *Computational Geometry: An Introduction* (Springer, Heidelberg, 1985; Mir, Moscow, 1989).

2021, **22** (4), 294–305. doi 10.26089/NumMet.v22r419

14. B. E. Alekseev and V. A. Talanov, *Graphs. Computational Models. Structures* (Lobachevsky State Univ. of Nizhni Novgorod, Nizhni Novgorod, 2005) [in Russian].
15. K. Paton, "An Algorithm for Finding a Fundamental Set of Cycles of a Graph," Commun. ACM **12** (9), 514–518 (1969).
16. E. M. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms. Theory and Practice* (Prentice-Hall, Englewood Cliffs, 1977; Mir, Moscow, 1980).
17. B. N. Ivanov, "Nesting Structures of the Isoline Field in the Gradient Filling Problem," Vychisl. Metody Programm. **7** (2), 30–40 (2006).

*Received*
*September 13, 2021*

*Accepted for publication*
*November 5, 2021*

**Information about the author**

*Boris N. Ivanov* — PhD., Associate Professor, Far Eastern Federal University, Institute of Mathematics and Computer Technologies, ulitsa Sukhanova 8, 690950, Vladivostok, Russia.