

Размер памяти для хранения упорядоченного корневого графа

Игорь Бурдонов <igor@ispras.ru>

Александр Косачев <kos@ispras.ru>

*Институт системного программирования РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.*

Аннотация. В статье рассматривается размер памяти, необходимый и достаточный для хранения неориентированных корневых упорядоченных связных графов как нумерованных, так и ненумерованных. Введение содержит основные определения и постановку задачи. Граф корневой, если одна из вершин выделена и названа корнем. Граф упорядоченный, если для каждой вершины все инцидентные ей рёбра линейно упорядочены. Граф нумерованный, если все его вершины помечены различными идентификаторами, в частности, пронумерованы целыми числами от 0 до $n-1$, где n число вершин графа. Два неориентированных корневых упорядоченных графа G и G' считаются слабо изоморфными, если существует взаимно-однозначное соответствие вершин такое, что: 1) соответствующие вершины имеют одинаковые степени, 2) рёбра ab из графа G и $a'b'$ из графа G' , инцидентные соответствующим вершинам a и a' и имеющие в этих вершинах одинаковые номера, ведут в соответствующие вершины b и b' , 3) корни соответствуют друг другу. Для нумерованных графов при изоморфизме дополнительно требуется совпадение номеров соответствующих вершин. Графы рассматриваются с точностью до слабого изоморфизма. Показано, что память, необходимая и достаточная для хранения любого графа, имеет размер $\Theta(m \log n)$ для нумерованных графов, $\Theta(n + (m-n+1) \log n)$ для ненумерованных графов с числом вершин n и числом рёбер m , и $\Theta(n^2 \log n)$ для графов без кратных рёбер и петель с числом вершин n . Также показано, что память, достаточная для хранения последовательности рёбер длины $O(n)$ или остова графа, имеет размер $O(n \log(n\Delta))$ или $O(n \log \Delta)$, соответственно, где Δ максимальная степень вершины.

Ключевые слова: неориентированный граф, упорядоченный граф, нумерованный граф, корневой граф, представление графа, перечисление графов.

1. Введение

В задачах исследования графов часто возникает проблема определения размера памяти, необходимой для хранения графа. Понятно, что минимальный размер такой памяти определяется числом графов интересующего нас класса и представляет собой логарифм от этого числа. Однако проблема определения числа графов того или иного класса, называемая также проблемой перечисления графов, – это, как правило, достаточно трудная проблема и редко удаётся получить результаты, выражаемые простыми формулами.

На рис. 1 показаны виды графов, которые рассматриваются в данной статье. Будем обозначать: n – число вершин, m – число рёбер, Δ – максимальная степень вершины.

Говорят, что граф *нумерованный* (или помеченный), если все его вершины помечены различными идентификаторами; мы будем считать, что они пронумерованы числами от 0 до $n-1$. Граф *корневой*, если одна из его вершин выделена, такая вершина называется *корнем*.

По теореме Кэли число некорневых нумерованных деревьев равно n^{n-2} [1]. Поскольку корень может быть выбран n разными способами, число корневых нумерованных деревьев равно n^{n-1} . Известно число нумерованных графов без петель и кратных рёбер с заданным числом вершин n . Оно равно $2^{n(n-1)/2}$ [2]. Если допускаются петли, то число графов возрастает до $2^{n(n-1)/2+n} = 2^{n(n+1)/2}$.

Будем называть граф *упорядоченным*, если задано локальное упорядочение его рёбер: для каждой вершины a все инцидентные ей рёбра перенумерованы от 1 до $\Delta(a)$, где $\Delta(a)$ степень вершины a . В неориентированном упорядоченном графе каждое ребро получает два номера – по одному в каждом из своих концов; петля также получает два номера в инцидентной ей вершине.

И. Бурдонов, А. Косачев.

Размер памяти для хранения упорядоченного корневого графа.

Труды Института системного программирования РАН. Том 29, выпуск 2, 2017, стр. 7-26. ISSN 2220-6426 (Online), ISSN 2079-8156 (Print).

20 стр.

В корневом дереве рёбрам можно придать естественную ориентацию от корня: ребро ab получает ориентацию $a \rightarrow b$, если вершина a лежит на пути в дереве от корня до вершины b , т.е. b потомок a . В таком ориентированном дереве корень имеет полустепень захода 0, а некорневая вершина имеет полустепень захода 1. Когда мы ниже будем говорить об ориентированных рёбрах в корневом дереве, мы будем иметь в виду именно эту ориентацию.

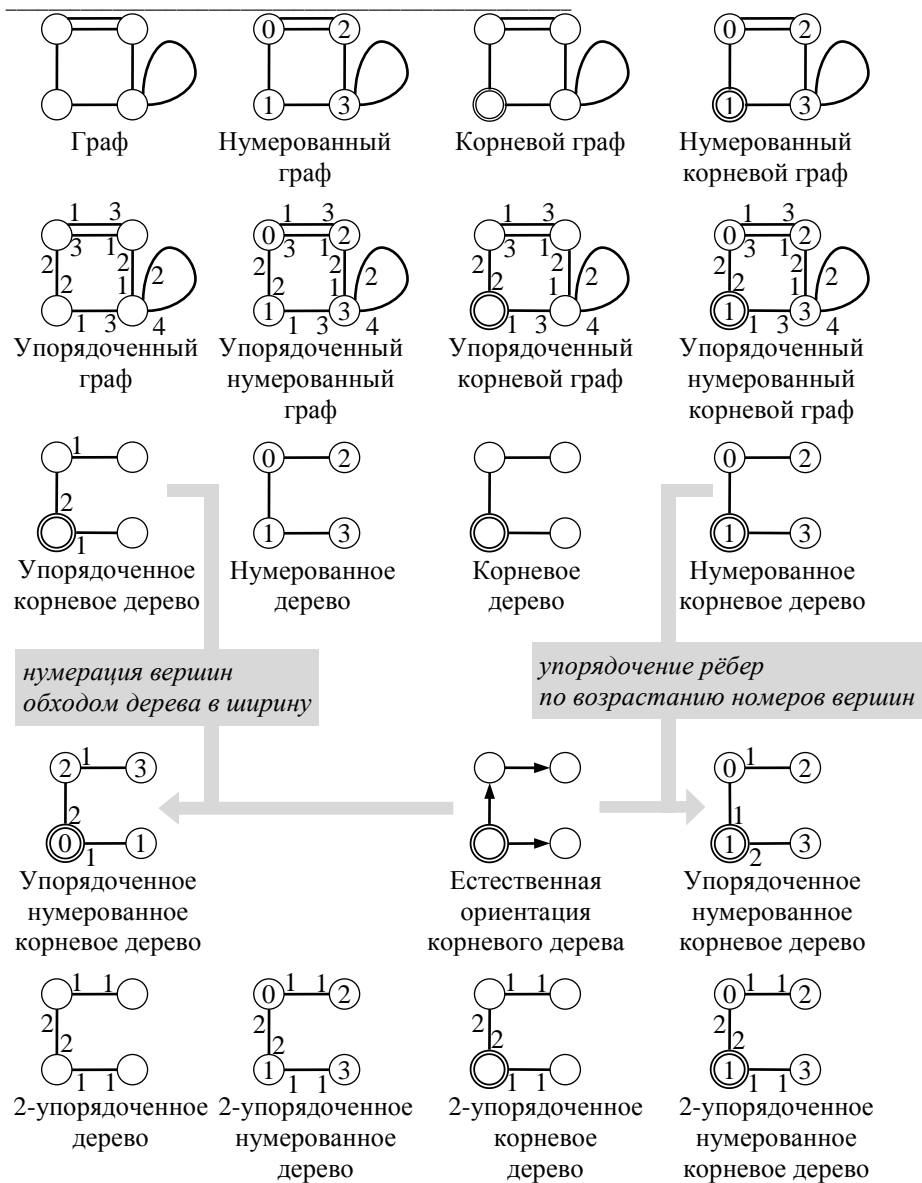


Рис. 1. Виды графов.

Корневое дерево называется *упорядоченным*, если для каждой вершины a её потомки, т.е. концы рёбер, выходящих из a при естественной ориентации рёбер, линейно упорядочены, т.е. пронумерованы от 1 до $\Delta(a)-1$. Заметим, что упорядоченное корневое дерево не является упорядоченным графом, если $n > 1$, поскольку в нем ребро $a \rightarrow b$ имеет номер только в вершине a , но не в вершине b . Однако если в вершине b ему присвоить любой номер i от 1 до $\Delta(b)$, а также увеличить на 1 каждый номер $j \geq i$ ребра, выходящего из b , то такое упорядоченное дерево станет упорядоченным графом. Мы будем называть его *2-упорядоченным* деревом. Понятно, что одному упорядоченному дереву соответствует, вообще говоря, множество 2-упорядоченных деревьев при разных выборах $i=1..\Delta(b)$.

Число упорядоченных корневых деревьев с n вершинами – это число Каталана (последовательность A000108 в OEIS [3], называемая также *Segner numbers*) $C_{n-1} = C_{2n-2}^{n-1}/n = (2n-2)!/((n-1)!n!)$ [4][5].

Нумерованное корневое дерево задаёт естественное упорядочение рёбер, выходящих из вершины при естественной ориентации рёбер корневого дерева, по возрастанию номеров концов рёбер. Упорядоченное корневое дерево, наоборот, задаёт естественную нумерацию вершин обходом дерева в ширину. Но при этом получается не любое нумерованное корневое дерево: номер 0 всегда получает корень, номер 1 – вершина, которая отлична от корня и соединена с корнем ребром с минимальным номером в корне, и т.д. В общем, как видно из сравнения чисел n^{n-1} и C_{n-1} , нумерованных деревьев больше, чем упорядоченных.

Будем говорить, что два неориентированных упорядоченных графа G и G' *сильно изоморфны*, если существует взаимно-однозначное соответствие

вершин такое, что: 1) соответствующие вершины имеют одинаковые степени, 2) рёбра ab из графа G и $a'b'$ из графа G' , инцидентные соответствующим вершинам a и a' и имеющие в этих вершинах одинаковые номера $i=i'$, ведут в соответствующие вершины b и b' и имеют в них одинаковые номера $j=j'$. При слабом изоморфизме не требуется совпадение вторых номеров рёбер, т.е. допускается $j \neq j'$. Для корневых упорядоченных графов при изоморфизме будем дополнительно требовать, чтобы корни соответствовали друг другу. Для нумерованных упорядоченных графов при изоморфизме будем дополнительно требовать, чтобы номера соответствующих вершин совпадали. При отсутствии кратных рёбер, в частности, для деревьев, оба изоморфизма совпадают. Действительно, в вершине b' должно быть ребро с номером j , поскольку ребро с таким номером есть в вершине b , b соответствует b' и, следовательно, степени вершин совпадают. Если бы было $j \neq j'$, то ребро с номером j должно было бы вести из b' в вершину $c' \neq a'$, поскольку из b' в a' уже ведет ребро с другим номером j' , а кратных рёбер нет. Следовательно, вершины b и b' соответствуют друг другу, но ребро с номером j ведет из b в a и из b' в $c' \neq a'$, что противоречит изоморфизму. При наличии кратных рёбер сильный и слабый изоморфизм различаются, пример приведён на рис. 2.

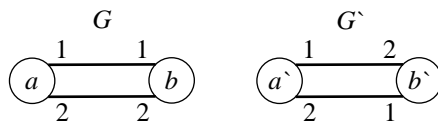


Рис. 2. Граф G слабо (но не сильно!) изоморфен графу G' .

Если \mathbf{G} некоторый класс неориентированных упорядоченных графов, то через $N(\mathbf{G})$ обозначим число графов в нём с точностью до слабого изоморфизма.

В данной статье рассматриваются классы $\mathbf{G}_1(n,m)$ нумерованных и $\mathbf{G}_2(n,m)$ ненумерованных неориентированных упорядоченных корневых связных

графов с n вершинами и m рёбрами (петли и кратные рёбра допускаются), а также классы $G_1(n)$ нумерованных и $G_2(n)$ ненумерованных графов с n вершинами без кратных рёбер и петель. Мы дадим точную по порядку оценку размера памяти, необходимой и достаточной для хранения любого графа из этих классов с точностью до слабого изоморфизма: $N(G_1(n,m)) = \Theta(m \log n)$, $N(G_2(n,m)) = \Theta(n + (m-n+1) \log n)$, $N(G_1(n)) = N(G_2(n)) = \Theta(n^2 \log n)$.

2. Нумерованные графы

В этом разделе мы рассмотрим графы из класса $G_1(n,m)$: нумерованные неориентированные упорядоченные корневые связные графы с n вершинами и m рёбрами. Мы дадим оценку сверху $O(m \log n)$ для размера $l_1(n,m)$ памяти, достаточной для хранения графа из этого класса, и оценку снизу для числа $N_1(n,m) = N(G_1(n,m))$ графов этого класса с точностью до слабого изоморфизма, равную $\Omega(n^m)$, что даст оценку снизу $\Omega(m \log n)$ для размера $L_1(n,m)$ памяти, необходимой для хранения графа из этого класса. Поскольку эти оценки размера памяти совпадают, размер памяти, необходимой и достаточной для хранения графов из класса $G_1(n,m)$, равен $\Theta(m \log n)$.

Если граф нумерованный, мы для краткости будем говорить «вершина a » вместо «вершина с номером a ». Через $\Delta_G(a)$ будем обозначать степень вершины a в графе G . Для нумерованного упорядоченного корневого графа G будем обозначать: $root(G)$ – номер корня графа G ; (a,i) – ребро, инцидентное вершине a и имеющее в ней номер i ; $\delta_G(a,i)$ – другой конец ребра (a,i) ; $v_G(a,i)$ – номер ребра (a,i) в вершине $\delta_G(a,i)$. По определению $\delta_G(\delta_G(a,i), v_G(a,i)) = a$ и $v_G(\delta_G(a,i), v_G(a,i)) = i$. Если граф G подразумевается, то его указание в нижнем индексе будем опускать.

2.1. Представление графа в памяти

В этом подразделе мы опишем способ представления графа из класса $G_1(n, m)$ в памяти размером $l_1(n, m) = O(m \log n)$. Таких представлений может быть несколько, мы дадим одно из них (рис. 3).

Мы будем применять два представления целых неотрицательных чисел. Обозначим $r(x) = 1$, если $x = 0$, и $r(x) = \lfloor \log_2 x \rfloor + 1$, если $x > 0$, где $\lfloor z \rfloor$ обозначает ближайшее к z снизу целое число, а $\log_2 x$ – двоичный логарифм. Если число x – это любое число из интервала $[0, p]$, где p известно, то будем использовать двоичный код фиксированной длины $r(p)$. Такое представление для $x = \sum \{x_i \cdot 2^{r(p)-i} \mid i=1..r(p)\}$, где $x_i \in \{0, 1\}$, обозначим $D_p(x) = x_1, x_2, \dots, x_{r(p)}$. Оно занимает $r(p)$ бит памяти. Если число x любое, не ограниченное сверху известной величиной, то минимально возможная разрядность его двоичного кода равна $r(x)$. В этом случае в представлении числа x каждый его разряд будем предварять кодом 0, а в конце поместим код 1. Такое представление для $x = \sum \{x_i \cdot 2^{r(x)-i} \mid i=1..r(x)\}$, где $x_i \in \{0, 1\}$, обозначим $D(x) = 0, x_1, 0, x_2, \dots, 0, x_{r(x)}, 1$. Оно занимает $2r(x)+1$ бит памяти.

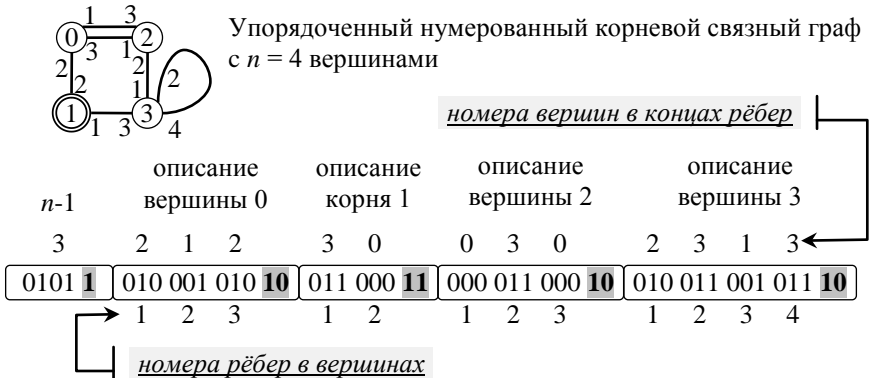


Рис. 3. Описание нумерованного графа.

Представление графа – это битовая последовательность:

$D(n-1)$, *Вершина*(0), *Вершина*(1),... *Вершина*($n-1$).

Здесь *Вершина*(a), где $a=0..n-1$, является описанием вершины a . Оно представляет собой битовую последовательность

Ребро($a,1$), *Ребро*($a,2$), ... *Ребро*($a,\Delta(a)$), *Ограничитель вершины*.

Здесь *Ребро*(a,i) – описание конца ребра (a,i), представляющее собой битовую последовательность $0,D_{n-1}(\delta(a,i))$. Это описание занимает $1+r(n-1)$ бит.

Ограничитель вершины – это двухбитовый код 10, если вершина с номером a – это не корень, и 11 для корня.

Заметим, что каждое ребро (a,i) (включая петли) описывается дважды: в вершине a , где указывается вершина $\delta(a,i)$, и в вершине $\delta(a,i)$, где указывается вершина a ; петля ($\delta(a,i) = a$) также дважды описывается, поскольку она тоже имеет два номера i и $v(a,i)$, но оба в вершине a . Тем самым, число описаний *Ребро*(a,i) равно $2m$.

При таком описании графа для любого ребра (a,i) мы можем по описанию *Вершина*(a) узнать вершину $\delta(a,i)$, в которую ведет это ребро. Однако, по описанию *Вершина*($\delta(a,i)$) мы не можем узнать, какой номер $v(a,i)$ это ребро имеет в вершине $\delta(a,i)$, но только подмножество номеров рёбер, одним из которых является номер $v(a,i)$. Это подмножество номеров j таких, что в описании *Вершина*($\delta(a,i)$) указана вершина a как другой конец ребра j : $\delta(\delta(a,i),j) = a$. Таким образом, описание определяет граф с точностью до слабого изоморфизма.

Заметим, что таким способом можно описать любой нумерованный упорядоченный корневой граф, а не только связный. Если граф не корневой,

то *Ограничитель вершины* может быть однобитовым кодом 1, что уменьшает описание на n бит.

При $n \geq 2$ описание графа занимает память размером

$$l_1(n, m) = 2r(n-1)+1 + 2m(1+r(n-1)) + 2n = 2r(n-1) + 2mr(n-1) + 2m + 2n + 1.$$

При $n=1$, имеем $l_1(1, m) = 2 \cdot 1 + 2m \cdot 1 + 2m + 2 \cdot 1 + 1 = 4m + 5$.

$$\begin{aligned} \text{При } n \geq 2 \text{ имеем } l_1(n, m) &= 2(\lfloor lb(n-1) \rfloor + 1) + 2m(\lfloor lb(n-1) \rfloor + 1) + 2m + 2n + 1 = \\ &= 2\lfloor lb(n-1) \rfloor + 2m\lfloor lb(n-1) \rfloor + 4m + 2n + 3. \end{aligned}$$

В частности, $l_1(2, m) = 2 \cdot 0 + 2m \cdot 0 + 4m + 2 \cdot 2 + 3 = 4m + 7 > l_1(1, m)$.

Поскольку $lb(n-1) < lbn$ и $\lfloor x \rfloor \leq x$, имеем $l_1(n, m) < 2lbn + 2mlbn + 4m + 2n + 3$.

Поскольку в связном графе $n \leq m+1$, имеем

$$l_1(n, m) < 2lbn + 2mlbn + 4m + 2(m+1) + 3 = 2lbn + 2mlbn + 6m + 5.$$

Для $m \geq 1$ имеем $l_1(n, m) \leq 2mlbn + 2mlbn + 6m + 5m = 4mlbn + 11m$.

Для $n \geq 2$ имеем $1 \leq lbn$ и поэтому $l_1(n, m) \leq 4mlbn + 11mlbn = 15mlbn$.

Итак, для $m \geq 1$ и $n \geq 2$ имеем $l_1(n, m) \leq 15mlbn$, т.е. $l_1(n, m) = O(m \log n)$.

2.2. Хранение графов с меньшим числом вершин и/или рёбер

Функция $l_1(n, m)$ монотонно неубывающая по каждой переменной n и m :

$l_1(1, m) < l_1(1, m+1)$, $l_1(1, m) < l_1(2, m)$ и, если $n \geq 2$, то $l_1(n, m) \leq l_1(n, m+1)$ и

$l_1(n, m) \leq l_1(n+1, m)$. Поэтому память размером $l_1(n, m)$ достаточна для хранения

любого нумерованного упорядоченного корневого графа с числом вершин

$n \leq n$ и числом рёбер $m \leq m$.

2.3. Число графов

В этом подразделе мы оценим снизу число $N_1(n, m)$ графов в классе $G_1(n, m)$ с

точностью до слабого изоморфизма и, соответственно, размер $L_1(n, m)$ памяти,

необходимой для хранения таких графов. Мы построим класс графов $G_1^2(n,m) \subseteq G_1(n,m)$ и подсчитаем число графов в нем (рис. 4).

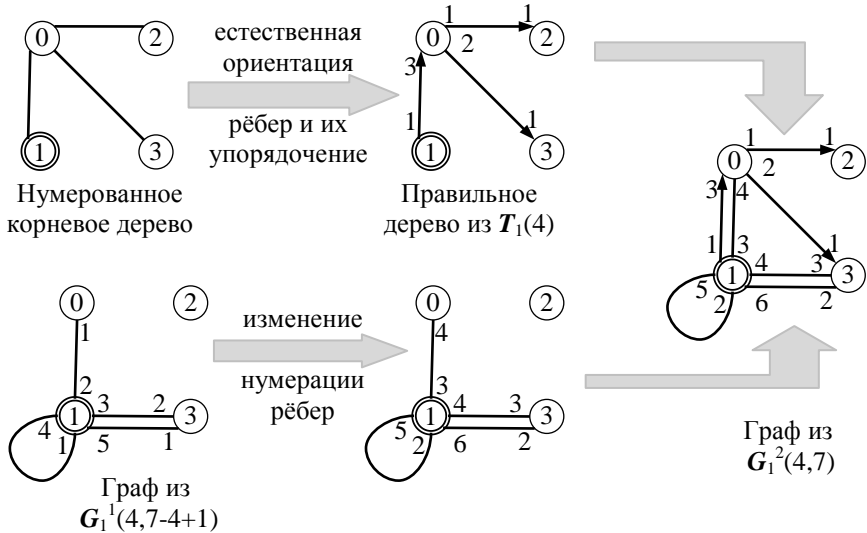


Рис. 4. Построение класса графов $G_1^2(n,m)$ для $n=4$ и $m=7$.

Правильным деревом будем называть 2-упорядоченное нумерованное корневое дерево, в котором при естественной ориентации рёбер ребро $b \rightarrow a$, входящее в вершину a , имеет в вершине a номер $\Delta(a)$, а последовательности рёбер, выходящих из вершины a , в порядке их нумерации в вершине a от 1 до $\Delta(a)-1$ соответствует возрастающая последовательность номеров конечных вершин этих рёбер: $\forall a=0..n-1 \forall i,j=1..\Delta(a)-1 (i < j \Rightarrow \delta(i) < \delta(j))$.

Правильное дерево однозначно получается из заданного нумерованного корневого дерева преобразованием f_1 , которое нумерует рёбра в вершинах в соответствии с номерами конечных вершин рёбер при естественной ориентации рёбер. А именно: для каждой некорневой вершины a все рёбра,

выходящие из a , т.е. рёбра вида $a \rightarrow b$, нумеруются от 1 до $\Delta(a)-1$ по порядку возрастания номеров концов c этих рёбер, а входящее ребро $c \rightarrow a$ получает в вершине a номер $\Delta(a)$; в корне e нет входящего ребра, а все выходящие рёбра перенумерованы от 1 до $\Delta(e)$ также по порядку возрастания номеров концов этих рёбер.

Очевидно, что операция удаления из правильного дерева нумерации рёбер в вершинах определена для любого правильного дерева и обратна к преобразованию f_1 . Поэтому отображение f_1 биективно. Следовательно, число правильных деревьев с n вершинами равно числу нумерованных корневых деревьев с n вершинами. Множество правильных деревьев с n вершинами обозначим $T_1(n)$. Тогда по теореме Кэли $T_1(n) = n^{n-1}$.

Рассмотрим следующую операцию добавления к правильному дереву $T \in T_1(n)$ графа $G \in G_1(n, m-n+1)$. Сначала изменим в G нумерацию рёбер: для каждого ребра (a, i) его номер i увеличим на $\Delta_T(a)$. После этого добавим рёбра графа G к остову T , используя взаимно-однозначное соответствие вершин T и G по номерам вершин. В качестве корня выберем корень дерева T . Результат этой операции обозначим через $T+G$. Формально:

$$root(T+G) = root(T), \forall a=0..n-1 \Delta_{T+G}(a) = \Delta_T(a) + \Delta_G(a),$$

$$\forall i=1.. \Delta_T(a) \delta_{T+G}(a, i) = \delta_T(a, i) \ \& \ \forall_{T+G}(a, i) = \nu_T(a, i),$$

$$\forall i=\Delta_T(a)+1.. \Delta_{T+G}(a) \delta_{T+G}(a, i) = \delta_G(a, i-\Delta_T(a)) \ \& \ \nu_{T+G}(a, i) = \nu_G(a, i-\Delta_T(a)).$$

Если T изоморфно T' , а G слабо изоморфно G' , то

$$root(T+G) = root(T) = root(T') = root(T'+G),$$

$$\forall a=0..n-1 \Delta_{T+G}(a) = \Delta_{T'}(a) \ \& \ \Delta_G(a) = \Delta_{G'}(a) \ \&$$

$$\forall i=1.. \Delta_T(a) \delta_T(a, i) = \delta_{T'}(a, i) \ \& \ \forall i=1.. \Delta_G(a) \delta_G(a, i) = \delta_{G'}(a, i).$$

Отсюда непосредственно следует, что $T+G$ слабо изоморфно $T'+G'$.

Если G не слабо изоморфно G^{\setminus} , то

$$\exists a=0..n-1 \Delta_G(a) \neq \Delta_{G^{\setminus}}(a) \vee \Delta_G(a) = \Delta_{G^{\setminus}}(a) \ \& \ \exists i=1..\Delta_G(a) \ \delta_G(a,i) \neq \delta_{G^{\setminus}}(a,i).$$

Поэтому, если T изоморфно T^{\setminus} , а G не слабо изоморфно G^{\setminus} , то $T+G$ не слабо изоморфно $T^{\setminus}+G^{\setminus}$.

Если T не изоморфно T^{\setminus} , то 1) $root(T) \neq root(T^{\setminus}) \vee \exists a=0..n-1$

$$2) \Delta_T(a) \neq \Delta_{T^{\setminus}}(a) \vee 3) \Delta_T(a) = \Delta_{T^{\setminus}}(a) \ \& \ \exists i=1..\Delta_T(a) \ \delta_T(a,i) \neq \delta_{T^{\setminus}}(a,i).$$

Если выполнено условие 1, то $root(T+G) \neq root(T^{\setminus}+G^{\setminus})$, т.е. графы $T+G$ и $T^{\setminus}+G^{\setminus}$ не слабо изоморфны. Если для некоторого $a=0..n-1$ выполнено условие 3, то $\delta_{T+G}(a,i) = \delta_T(a,i) \neq \delta_{T^{\setminus}}(a,i) = \delta_{T^{\setminus}+G^{\setminus}}(a,i)$, т.е. графы $T+G$ и $T^{\setminus}+G^{\setminus}$ не слабо изоморфны. Пусть для любого $a=0..n-1$ условие 3 не выполнено, но для некоторого $a=0..n-1$ выполнено условие 2: $\Delta_T(a) \neq \Delta_{T^{\setminus}}(a)$. Поскольку сумма степеней вершин дерева равна удвоенному числу его рёбер, т.е. $2(n-1)$, то хотя бы одна из таких вершин a не является корнем. Тогда, поскольку условие 3 не выполнено, в деревьях T и T^{\setminus} существует общий путь ненулевой длины от корня до некоторой вершины a , в котором степени всех вершин, кроме a , совпадают в T и T^{\setminus} , а в a не совпадают. Пусть для определенности $\Delta_T(a) < \Delta_{T^{\setminus}}(a)$. Рассмотрим ребро $(a, \Delta_T(a))$ в T и в $T^{\setminus}+G^{\setminus}$. Поскольку дерево T правильное, это ребро $b \rightarrow a$ входящее в вершину a , т.е. $\delta_T(a, \Delta_T(a)) = b$. Поскольку $\Delta_T(a) < \Delta_{T^{\setminus}}(a)$, в T^{\setminus} тоже есть ребро $(a, \Delta_T(a))$, но там оно не может заканчиваться в b , поскольку дерево T^{\setminus} тоже правильное и в b должно заканчиваться ребро $(a, \Delta_{T^{\setminus}}(a))$. Поскольку $\Delta_T(a) \leq \Delta_T(a) < \Delta_{T^{\setminus}}(a)$, имеем $\delta_{T+G}(a, \Delta_T(a)) = \delta_T(a, \Delta_T(a)) \neq \delta_{T^{\setminus}}(a, \Delta_T(a)) = \delta_{T^{\setminus}+G^{\setminus}}(a, \Delta_T(a))$, т.е. графы $T+G$ и $T^{\setminus}+G^{\setminus}$ не слабо изоморфны.

Итак, мы доказали, что $T+G$ и $T^{\setminus}+G^{\setminus}$ слабо изоморфны тогда и только тогда, когда T изоморфно T^{\setminus} , а G слабо изоморфно G^{\setminus} .

Теперь рассмотрим класс $G_1^1(n, m-n+1)$ графов, в которых все рёбра инцидентны корню, имеющему номер 0. Каждое из $m-n+1$ рёбер, инцидентных корню, можно провести независимым образом в любую из n вершин. Следовательно, $N(G_1^1(n, m-n+1)) = n^{m-n+1}$. Для ребра, ведущего из корня a в вершину $b \neq a$, указывается только его номер в a и вершина b . После определения всех таких кратных рёбер ab они должны быть упорядочены в вершине b , но способ упорядочивания мы не фиксируем. Для петли с номером i в корне её другой номер $v(a, i)$ – это один из номеров $j \neq i$, выделенных в корне для петель, но какой именно номер, не указывается. Поэтому таким способом мы определяем нумерованные упорядоченные корневые графы с точностью до слабого изоморфизма.

Обозначим: $G_1^2(n, m) = T_1(n) + G_1^1(n, m-n+1) = \{T + G | T \in T_1(n) \ \& \ G \in G_1^1(n, m-n+1)\}$.

В силу доказанного, число графов в классе $G_1^2(n, m)$ с точностью до слабого изоморфизма равно произведению числа деревьев в классе $T_1(n)$ с точностью до изоморфизма и числа графов в классе $G_1^1(n, m-n+1)$ с точностью до слабого изоморфизма: $N_1(n, m) \geq N(G_1^2(n, m)) = N(T_1(n))N(G_1^1(n, m-n+1)) = n^{n-1}n^{m-n+1} = n^m$.

Тем самым, для хранения графов из класса $G_1(n, m)$ требуется память $L_1(n, m) = \lfloor \lg(N_1(n, m)-1) \rfloor + 1 \geq \lfloor \lg(n^m-1) \rfloor + 1$. Поскольку $\lfloor x \rfloor \geq x-1$, и $n^m-1 \geq n^{m-1}$ при $n \geq 2$ и $m \geq 1$, имеем $L_1(n, m) \geq \lg(n^m-1)-1+1 \geq \lg n^{m-1} = (m-1)\lg n$, т.е. $L_1(n, m) = \Omega(m \log n)$.

3. Ненумерованные графы

В этом разделе мы рассмотрим графы из класса $G_2(n, m)$: ненумерованные неориентированные упорядоченные корневые связные графы с n вершинами и m рёбрами. Поскольку графы связные, $n \leq m+1$ и $m-n+1 \geq 0$. Мы дадим оценку сверху $O(n+(m-n+1)\log n)$ для размера $l_2(n, m)$ памяти, достаточной для хранения графа из этого класса, и оценку снизу для числа $N_2(n, m) = N(G_2(n, m))$

графов этого класса с точностью до слабого изоморфизма, равную $\Omega(2^n n^{m-n+1})$, что даст оценку снизу $\Omega(n+(m-n+1)\log n)$ для размера $L_2(n,m)$ памяти, необходимой для хранения графа из этого класса. Поскольку эти оценки совпадают, размер памяти, необходимой и достаточной для хранения графов из класса $G_2(n,m)$, равен $\Theta(n+(m-n+1)\log n)$.

3.1. Представление графа в памяти

В этом подразделе мы опишем способ представления графа из класса $G_2(n,m)$ в памяти размером $l_2(n,m) = O(n+(m-n+1)\log n)$. Таких представлений может быть несколько, мы дадим одно из них (рис. 5). Это представление похоже на представление графа из подраздела 2.1 для нумерованных графов. Для этого используется естественная нумерация вершин ненумерованного упорядоченного корневого графа, основанная на обходе графа в ширину. Опишем эту процедуру нумерации вершин.

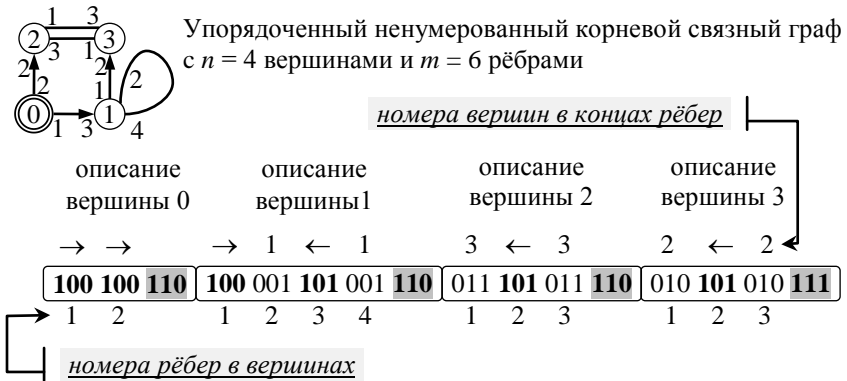


Рис. 5. Описание ненумерованного упорядоченного графа (показана естественная нумерация вершин).

Корень получает номер 0. Выбирается первое ребро, инцидентное корню и не являющееся петлей; его конец получает номер 1. Далее выбирается

следующее ребро, инцидентное корню и не ведущее в вершины 0 и 1; его конец получает номер 2. И так далее. Когда исчерпаются все рёбра, инцидентные корню, аналогичная процедура выполняется для вершины 1.

В общем, на каждом шаге у нас определены номера вершин $0, \dots, a^`$ и перебраны все рёбра, инцидентные вершинам $0, \dots, a-1$, где $a < a^`$, а для вершины a перебраны рёбра с номерами $1, \dots, i-1$. Если $i \leq \Delta(a)$, то для вершины a рассматривается ребро с номером i . Если оно ведет в одну из вершин, уже получивших номера, т.е. в вершины $0, \dots, a^`$, то переходим к следующему ребру $i+1$. В противном случае конец ребра с номером i получает номер $a^`+1$. Когда все рёбра, инцидентные вершине с номером a , будут перебраны, т.е. когда $i > \Delta(a)$, переходим к следующей нумерованной вершине $a+1$, если $a < a^`$. Если же $a = a^`$, то процедура нумерации вершин закончена.

В получившемся нумерованном графе однозначно выделяется корневой остов: ребро (a, i) принадлежит остову тогда и только тогда, когда вершина $\delta(a, i)$ получает свой номер при просмотре именно этого ребра в процедуре нумерации. Формально это означает, что из вершин, номера которых меньше a , ни одно ребро не ведет в вершину $\delta(a, i)$, т.е. $c < a \Rightarrow \forall j=1.. \Delta(c) \delta(c, j) \neq \delta(a, i)$, и рёбра, инцидентные вершине a и имеющие в ней номера меньше i , также не ведут в вершину $\delta(a, i)$, т.е. $\forall j=1..i-1 \delta(a, j) \neq \delta(a, i)$. Заметим, что таким образом выделяется корневой остов только при естественной нумерации вершин, описанной выше.

Теперь мы уже можем использовать представление графа, описанного в подразделе 2.1. Отличие будет в том, что для рёбер выделенного остова мы не будем указывать номера вершин. Пусть (a, i) ребро остова. Тогда описание *Ребро* (a, i) будет не последовательностью $0, D_{n-1}(\delta(a, i))$, а трёхбитовым кодом 100, а описание *Ребро* $(\delta(a, i), v(a, i))$ будет не последовательностью $0, D_{n-1}(a)$, а

кодом 101. Тем самым, описание ребра остова будет занимать 3 бита, число таких описаний равно $2(n-1)$. Описание не остовного ребра, по-прежнему, занимает $1+r(n-1)$ бит; число таких описаний равно $2(m-n+1)$. Ограничителем вершин будет код 110, занимающий также 3 бита. Специальная разновидность ограничителя для корневой вершины не требуется, поскольку корень имеет фиксированный номер 0. Зато мы можем использовать трёхбитовый код 111 вместо кода 110 для ограничителя последней вершины в списке вершин и, тем самым, не указывать число вершин.

В итоге описание графа занимает память размером

$$l_2(n, m) = 2(n-1) \cdot 3 + 2(m-n+1)(1+r(n-1)) + 3n = 2(m-n+1)(1+r(n-1)) + 9n - 6.$$

При $n=1$, имеем $l_2(1, m) = 2(m-1+1) \cdot (1+1) + 9 \cdot 1 - 6 = 4m + 6$.

При $n \geq 2$ имеем $l_2(n, m) = 2(m-n+1)(1 + \lfloor \lg(n-1) \rfloor + 1) + 9n - 6 =$
 $= 2(m-n+1)(\lfloor \lg(n-1) \rfloor + 2) + 9n - 6$.

В частности, $l_2(2, m) = 2(m-2+1)(0+2) + 9 \cdot 2 - 6 = 4m + 8 > l_2(1, m)$.

Поскольку $\lg(n-1) < \lg n$ и $\lfloor x \rfloor \leq x$, имеем

$$l_2(n, m) < 2(m-n+1)(\lg n + 2) + 9n - 6.$$

Для $n \geq 2$ имеем $1 \leq \lg n$ и, поскольку также $-6 < 0$, получаем

$$l_2(n, m) < 2(m-n+1)(\lg n + 2\lg n) + 9n = 6(m-n+1)\lg n + 9n.$$

Поскольку в связном графе $m-n+1 \geq 0$, имеем $l_2(n, m) < 9((m-n+1)\lg n + n)$.

Итак, для $n \geq 2$ имеем $l_2(n, m) < 9((m-n+1)\lg n + n)$.

Следовательно, $l_2(n, m) = O(n + (m-n+1)\log n)$.

3.2. Хранение графов с меньшим числом вершин и/или рёбер

Для связного графа, когда $m-n+1 \geq 0$, функция $l_2(n, m)$ монотонно неубывающая по m : $l_2(n, m) \leq l_2(n, m+1)$. Также $l_2(1, m) < l_2(2, m)$, но, в отличие от

функции $l_1(n,m)$, функция $l_2(n,m)$ не является монотонно неубывающей по n при $n \geq 2$. Например,

$$l_2(9,9) = 2(9-9+1)(\lfloor lb(9-1) \rfloor + 2) + 9 \cdot 9 - 6 = 2 \cdot 1 \cdot (3+2) + 75 = 85, \text{ но}$$

$$l_2(10,9) = 2(9-10+1)(\lfloor lb(10-1) \rfloor + 2) + 9 \cdot 10 - 6 = 84.$$

Поэтому для того чтобы размер памяти был достаточен для хранения графов с меньшим или равным числом вершин и/или рёбер, т.е. графов из объединения $\cup \{G_2(n^{\wedge}, m^{\wedge}) \mid 2 \leq n^{\wedge} \leq n \ \& \ 0 \leq m^{\wedge} \leq m \ \& \ n^{\wedge} \leq m^{\wedge} + 1\}$ при $n \leq m+1$, нужно найти такое число вершин n^{\wedge} , на котором функция $l_2(n^{\wedge}, m)$ достигает максимума при заданном m и в интервале $2 \leq n^{\wedge} \leq n$. Искомый размер памяти равен $l_2(n^{\wedge}, m)$.

Как найти n^{\wedge} ? Обозначим $x = lb(n^{\wedge}-1)$. Тогда $n^{\wedge} = 2^x + 1$. Если n^{\wedge} целое число в интервале $2 \leq n^{\wedge} \leq n$, то x вещественное число в интервале $0 \leq x \leq lb(n-1)$.

Подставляя $2^x + 1$ вместо n^{\wedge} , имеем $l_2(n^{\wedge}, m) = 2(m-2^x)(\lfloor x \rfloor + 2) + 9(2^x + 1) - 6$.

Обозначим $l_2^{\wedge}(x, m) = 2(m-2^x)(x+2) + 9(2^x + 1) - 6$. Обозначим через x^{\wedge} значение x ,

при котором функция $l_2^{\wedge}(x, m)$ на интервале $0 \leq x \leq lb(n-1)$ принимает максимум. Программа Maple2015 и программа решения уравнений на сайте

«Решение математики онлайн» [7] дают одно решение:

$$x^{\wedge} = (W((m+1)e^{2^{1/2}/8}) - 1 + 2.5 \ln 2) / \ln 2, \text{ где } W \text{ функция Ламберта. Значение } 2^{x^{\wedge}} + 1$$

не обязательное целое, но $n_1 = \max\{\lfloor 2^{x^{\wedge}} + 1 \rfloor, 2\} \leq 2^{x^{\wedge}} + 1 \leq \min\{\lceil 2^{x^{\wedge}} + 1 \rceil, n\} = n_2$.

Тогда искомое значение n^{\wedge} есть n_1 или n_2 в зависимости от того, на каком из этих двух значений n_1 или n_2 функция $l_2(n^{\wedge}, m)$ принимает максимум: если

$$l_2(n_1, m) > l_2(n_2, m), \text{ то } n^{\wedge} = n_1, \text{ иначе } n^{\wedge} = n_2.$$

3.3. Число графов

В этом подразделе мы оценим снизу число $N_2(n,m)$ графов в классе $G_2(n,m)$ с точностью до слабого изоморфизма и, соответственно, размер $L_2(n,m)$ памяти, необходимой для хранения таких графов.

Упорядоченность графа, заданного с точностью до слабого изоморфизма, однозначно определяет естественную нумерацию вершин, основанную на обходе графа в ширину. Поэтому любая нумерация, задающая нумерованный упорядоченный граф из $G_1(n, m)$, есть, фактически, перенумерация графа из $G_2(n, m)$. Тем самым, $N_1(n, m) = N_2(n, m) \cdot n!$, где $n!$ – число различных нумераций вершин при фиксированном линейном порядке вершин. Поскольку $N_1(n, m) \geq n^m$, имеем $N_2(n, m) \geq n^m/n! \geq n^m e^n / (n+1)^{n+1}$. При $n \geq 35$ имеем $(n+1)^{n+1} \leq n^{n+2}$ и поэтому $N_2(n, m) \geq n^m e^n / n^{n+2} = n^{m-n-2} e^n$. Отсюда $L_2(n, m) = \lfloor \lg(N_2(n, m) - 1) \rfloor + 1 = \lfloor \lg(n^{m-n-2} e^n - 1) \rfloor + 1 \geq \lfloor \lg(n^{m-n-2} e^{n-1}) \rfloor + 1 \geq \lg(n^{m-n-2} e^{n-1}) - 1 + 1 = (m-n-2)\lg n + (n-1)\lg e = (m-n+1)\lg n - 3\lg n + (n-1)\lg e \geq (m-n+1)\lg n + n/2 \geq (1/2)((m-n+1)\lg n + n)$. Поэтому $L_2(n, m) = \Omega(n + (m-n+1)\lg n)$.

Мы также докажем это утверждение конструктивно, построив класс графов $G_2^2(n, m) \subseteq G_2(n, m)$ и подсчитав число графов в нем (рис. 6).

Как сказано во введении, число упорядоченных корневых деревьев с n вершинами – это число Каталана: $C_{n-1} = C_{2n-2}^{n-1}/n = (2n-2)!/((n-1)n!)$.

В [6] доказана оценка снизу: $C_{n-1} > 2^{2n-3}(4n-5)^{1/2}/((n-1)n\pi^{1/2})$.

Обозначим: $A(n) = 2^{n-3}(4n-5)^{1/2}$, $B(n) = (n-1)n\pi^{1/2}$. Тогда $C_{n-1} > 2^n A(n)/B(n)$.

Функция $A(n)$ при $n \geq 2$ увеличивается больше чем вдвое при увеличении аргумента на 1: $A(n+1) > 2A(n)$. Действительно, $A(n+1) = 2^{n+1-3}(4(n+1)-5)^{1/2} = 2 \cdot 2^{n-3}(4(n+1)-5)^{1/2} > 2 \cdot 2^{n-3}(4n-5)^{1/2} = 2A(n)$.

Функция $B(n)$ при $n \geq 3$ увеличивается не более чем вдвое при увеличении аргумента на 1: $B(n+1) \leq 2B(n)$. Действительно, $n \geq 3$ влечет $2n-2 \geq n+1$, что при $n > 0$ влечет $2n(n-1) \geq n(n+1)$, что влечет $2B(n) = 2n(n-1)\pi^{1/2} \geq n(n+1)\pi^{1/2} = B(n+1)$.

И. Бурдонов, А. Косачев.

Размер памяти для хранения упорядоченного корневого графа.

Труды Института системного программирования РАН. Том 29, выпуск 2, 2017, стр. 7-26. ISSN 2220-6426 (Online), ISSN 2079-8156 (Print).

20 стр.

Тем самым, при $n \geq 3$ имеем $A(n+1)/B(n+1) > 2A(n)/2B(n)$, т.е. функция $A(n)/B(n)$ монотонно увеличивается.

При $n=7$ имеем $A(7)/B(7) = 2^{7-3}(4 \cdot 7 - 5)^{1/2}/(7-1) \cdot 7\pi^{1/2} = 8 \cdot 23^{1/2}/21\pi^{1/2} > 1$.

Следовательно, при $n \geq 7$ имеем $C_{n-1} > 2^n A(n)/B(n) > 2^n$.

Упорядоченное дерево однозначным образом превращается в правильное дерево (см. подраздел 2.3) следующим преобразованием f_2 . Сначала, используя естественную ориентацию корневого дерева, каждому ребру, входящему в вершину a , присвоим в этой вершине номер $\Delta(a)$. Напомним, что при естественной ориентации корневого дерева в каждую вершину, кроме корня, входит ровно одно ребро. Мы получим 2-упорядоченное дерево, в котором ребро имеет два номера – по одному в каждом из его концов. Затем выполним естественную нумерацию вершин, основанную на обходе дерева в ширину, которую мы применяли в предыдущем подразделе 3.1. После такой нумерации дерево становится правильным. Множество полученных таким образом правильных деревьев обозначим через $T_2(n)$.

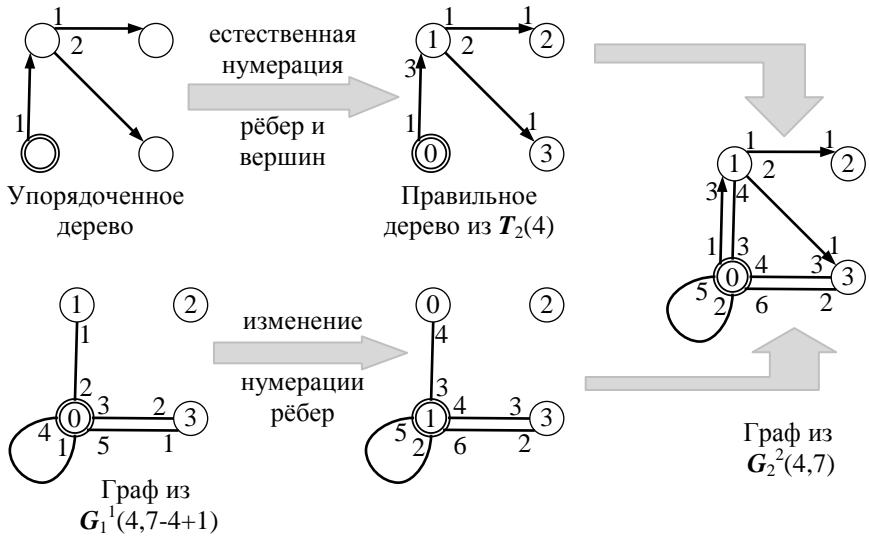


Рис. 6. Построение класса графов $G_2^2(n, m)$ для $n=4$ и $m=7$.

Очевидно, что если для упорядоченного корневого дерева T из правильного дерева $f_2(T)$ удалить нумерацию вершин и добавленные номера рёбер $\Delta(a)$, то получится исходное дерево T . Это означает, что преобразование f_2 является инъекцией. Поэтому $N(T_2(n)) = C_{n-1}$.

Заметим, что f_2 не сюръективно, например, в дереве $f_2(T)$ корень всегда имеет номер 0, что не обязательно в произвольном правильном дереве. Иными словами, дерево из $T_2(n)$ – это не любое правильное дерево из $T_1(n)$. Дерево из $T_1(n)$ биективно строится из нумерованного корневого дерева, а дерево из $T_2(n)$ биективно строится из упорядоченного дерева; нумерованных корневых деревьев больше, чем упорядоченных деревьев. Поэтому $T_2(n) \subset T_1(n)$.

Теперь рассмотрим операцию «+» добавления к дереву $T \in T_2(n)$ графа $G \in G_1^1(n, m-n+1)$, которую в подразделе 2.3 мы применяли для добавления графа G к правильному дереву из класса $T_1(n)$. Поскольку выше мы доказали,

что $T+G$ и $T^+G^`$ слабо изоморфны тогда и только тогда, когда T изоморфно $T^`$, а G слабо изоморфно $G^`$, в графе $T+G$ есть только один правильный остов T . Для дерева $T \in \mathcal{T}_2(n)$ его нумерация и, следовательно, нумерация графа $T+G$, однозначно определяется исходным упорядоченным корневым деревом с помощью преобразования f_2 . Если удалить нумерацию вершин и добавленные номера рёбер $\Delta(a)$, то из множества графов $\mathcal{T}_2(n)+\mathcal{G}_1^1(n,m-n+1)$ получится равномощное множество графов, которое обозначим $\mathcal{G}_2^2(n,m)$. Очевидно, $\mathcal{G}_2^2(n,m) \subset \mathcal{G}_1^2(n,m)$. Аналогично тому, как в подразделе 2.3 $N_1(n,m) \geq N(\mathcal{G}_1^2(n,m)) = N(\mathcal{T}_1(n))N(\mathcal{G}_1^1(n,m-n+1))$, имеем $N_2(n,m) \geq N(\mathcal{G}_2^2(n,m)) = N(\mathcal{T}_2(n))N(\mathcal{G}_1^1(n,m-n+1)) = C_{n-1}n^{m-n+1} > 2^n n^{m-n+1}$.

Отсюда следует, что для хранения графов из класса $\mathcal{G}_2(n,m)$ требуется память $L_2(n,m) = \lfloor \lg(N_2(n,m)-1) \rfloor + 1 \geq \lfloor \lg(2^n n^{m-n+1} - 1) \rfloor + 1$. Поскольку $2^n n^{m-n+1} - 1 \geq 2^{n-1} n^{m-n+1}$ при $n \geq 1$ и $m-n+1 \geq 0$, а $\lfloor \lg x \rfloor \geq x-1$, имеем $L_2(n,m) \geq \lfloor \lg 2^{n-1} n^{m-n+1} \rfloor + 1 \geq (n-1) + (m-n+1) \lg n - 1 + 1 = (n-1) + (m-n+1) \lg n$, т.е. $L_2(n,m) = \Omega(n + (m-n+1) \log n)$.

4. Графы без кратных рёбер и петель

В графах без кратных рёбер и петель число рёбер ограничено сверху: $m \leq n(n-1)/2$. Для класса всех таких графов с n вершинами полученные выше оценки для графов с n вершинами и m рёбрами корректируются.

Обозначим: $\mathcal{G}_1(n)$ – класс нумерованных и $\mathcal{G}_2(n)$ – класс нумерованных неориентированных упорядоченных корневых связных графов с n вершинами без кратных рёбер и петель, $n^\wedge = n(n-1)/2$, $N_1(n) = N_1(\mathcal{G}_1(n))$, $N_2(n) = N_2(\mathcal{G}_2(n))$, $L_1(n) = \lfloor \lg(N_1(n)-1) \rfloor + 1$, $L_2(n) = \lfloor \lg(N_2(n)-1) \rfloor + 1$.

Функция $l_1(n,m)$ монотонно неубывающая по переменной n , поэтому память размером $l_1(n) = l_1(n, n^\wedge)$ достаточна для хранения описания любого графа из $\mathcal{G}_1(n)$. Хотя функция $l_2(n,m)$, вообще говоря, не является монотонно

неубывающей по n , нетрудно убедиться, что при $m = n^\wedge$ она монотонно неубывающая при $n \geq 2$. Поэтому при $n \geq 2$ память размером $l_2(n) = l_2(n, n^\wedge)$ достаточна для хранения описания любого графа из $\mathbf{G}_2(n)$.

Для $n \geq 3$ имеем $l_2(n) = l_2(n, n^\wedge) = (n^\wedge - n + 1)lbn + n \leq n^\wedge lbn = l_1(n, n^\wedge) = l_1(n)$. Также $l_1(n) = \mathbf{O}(n^2 \log n)$. Также $N_1(n) \geq N_2(n)$ и, следовательно, $L_1(n) \geq L_2(n)$.

Покажем, что $L_2(n) = \mathbf{\Omega}(n^2 \log n)$. Для этого рассмотрим класс $\mathbf{G}_2^1(n) \subset \mathbf{G}_2(n)$ полных графов (степени всех вершин одинаковы и равны $n-1$), в каждом из которых есть простой цикл рёбер длины n с нумерацией рёбер по циклу $(1,2), (1,2), \dots, (1,2)$. Для такого графа можно определить естественную нумерацию вершин: вершина, достижимая из корня по последовательности k рёбер с номерами 1, получает номер k . Это значит, что $\forall a=0..n-2 \delta(a,1) = a+1$, $\delta(n-1,1) = 0$, $\forall a=1..n-1 \delta(a,2) = a-1$, $\delta(0,2) = n-1$.

При отсутствии кратных рёбер слабый изоморфизм графов совпадает с сильным изоморфизмом. Граф из $\mathbf{G}_2^1(n)$ однозначно определяется отображением в каждой вершине a номера ребра $i > 2$ в номер конца ребра, т.е. функцией $\delta(a,i)$. Поскольку кратных рёбер нет, эта функция инъективна по номеру ребра i : $\forall a=0..n-1 \forall i,j=1..n-1 i \neq j \Rightarrow \delta(a,i) \neq \delta(a,j)$. В каждой вершине имеем $(n-3)!$ различных упорядочиваний рёбер с номерами больше 2, а всего получаем $N_2(n) = N_2(\mathbf{G}_2(n)) \geq N(\mathbf{G}_2^1(n)) = ((n-3)!)^n$ графов. Тем самым, $L_2(n) = \lfloor lb(N_2(n)) - 1 \rfloor + 1 \geq lb(N_2(n)) - 1 = lb(((n-3)!)^n) - 1 = n lb((n-3)!) - 1 \geq n lb((2\pi)^{0.5} (n-3)^{n-3+0.5} / e^{n-3}) - 1 > n^2 lbn$ при $n > 5$. Отсюда $L_2(n) = \mathbf{\Omega}(n^2 \log n)$.

Итак, память, необходимая и достаточная для хранения графа из класса (нумерованных или не нумерованных) упорядоченных корневых связных графов с n вершинами без кратных рёбер и петель, имеет размер $\mathbf{\Theta}(n^2 \log n)$.

5. Память для хранения последовательности $O(n)$ рёбер

В этом разделе мы исследуем вопрос о размере памяти для хранения последовательности рёбер длиной $O(n)$. Мы будем предполагать, что граф нумерованный из класса $G_1(n, m)$ или ненумерованный из класса $G_2(n, m)$. Во втором случае будем применять естественную нумерацию вершин, основанную на обходе графа в ширину и описанную в подразделе 3.1. Длину последовательности рёбер обозначим через h . Будем считать, что $n \geq 3$. Будем предполагать, что для некоторой константы $C > 0$ имеет место $0 < h \leq Cn$.

Последовательность рёбер будем представлять как битовую последовательность $Рёбро_1(1), Рёбро_1(2), \dots, Рёбро_1(h)$, 1. Здесь $Рёбро_1(i)$ – это битовая последовательность $D(a_i), D(b_i-1), D(c_i-1), D(d_i)$, представляющая четыре числа: a_i – это номер одной из вершин, которой инцидентно i -ое ребро, b_i – это номер этого ребра в вершине a_i , $c_i = v(a_i, b_i)$, $d_i = \delta(a_i, b_i)$. Здесь для числа x используется представление $D(x)$, поскольку число вершин n и максимальная степень вершины заранее неизвестны.

Поскольку $a_i \leq n-1$ и $n \geq 3$, имеем $D(a_i) \leq 2r(n-1)+1 = 2(\lfloor \lg(n-1) \rfloor + 1) + 1 < 2(\lfloor \lg(n-1) \rfloor + 1) + 1 < 2(\lg n + 1) + 1 = 2\lg n + 3$. Такую же оценку имеем для $D(d_i)$.

Поскольку $n \geq 3$, в связном графе $1 < \Delta$. Имеем $D(b_i-1) \leq 2r(\Delta-1)+1 \leq 2(\lfloor \lg(\Delta-1) \rfloor + 1) + 1 < 2(\lg(\Delta-1) + 1) + 1 < 2(\lg \Delta + 1) + 1 = 2\lg \Delta + 3$. Такую же оценку имеем для $D(c_i-1)$.

Тем самым представление последовательности рёбер занимает память размером $q(h) \leq 2h(2\lg n + 3 + 2\lg \Delta + 3) + 1 = 4h\lg n\Delta + 12h + 1 \leq 4Cn\lg n\Delta + 12Cn + 1$.

При $n \geq 3$ и $\Delta > 1$ имеем $\lg n\Delta \geq 1$. Поэтому $q(h) \leq 4Cn\lg n\Delta + 12Cn\lg n\Delta + n\lg n\Delta = (16C+1)n\lg n\Delta = O(n\log n\Delta)$.

Так как $\Delta \leq 2m$, имеем $q(h) \leq (16C+1)n(\lg nm + 1) < (32C+2)n\lg nm = O(n\log nm)$.

Обозначим верхнюю границу через $Q(n,m) = (32C+2)nlbnm$ и сравним ее с размером памяти, необходимым для описания графа: $L_1(n,m)$ для нумерованного графа и $L_2(n,m)$ для ненумерованного графа.

Поскольку $L_1(n,m) \geq mlbn$ и $L_2(n,m) \geq n+(m-n+1)lbn$, имеем:

$$Q(n,m)/L_1(n,m) \leq (32C+2)nlb(nm)/(m-1)lbn \text{ и}$$

$$Q(n,m)/L_2(n,m) \leq (32C+2)nlb(nm)/(n-1)+(m-n+1)lbn).$$

При фиксированном (но произвольном) $n > 1$ и $m \rightarrow \infty$ правые части неравенств стремятся к нулю. Это значит, что размер памяти, необходимый для хранения неориентированного упорядоченного корневого (нумерованного или ненумерованного) графа с n вершинами и достаточно большим по сравнению с n числом m рёбер, больше по порядку, чем размер памяти, достаточный для хранения произвольной последовательности рёбер этого графа длины $O(n)$.

Это, конечно, не означает, что всегда $Q(n,m) \leq mlbn$ и $Q(n,m) \leq n+(m-n+1)lbn$.

При малом по сравнению с n числе рёбер, например, $m = n$ и $n \geq 3$ имеем:

$$Q(n,n) = (32C+2)nlb(n-n) > (n-1)lbn > (n-1)+(n-n+1)lbn.$$

Для более детального анализа можно рассмотреть граф с n вершинами, состоящий только из одного простого цикла длиной n . Сравним размеры $l_1(n,n)$ и $l_2(n,n)$ описаний этого графа с размером $q(n)$ описания последовательности рёбер этого цикла. Имеем:

$$l_1(n,n) < 2lbn + 2nlbn + 4n + 2n + 3 = 2(n+1)lbn + 6n + 3,$$

$$l_1(n,n) = 2\lfloor lb(n-1) \rfloor + 2n\lfloor lb(n-1) \rfloor + 4n + 2n + 3 = 2(n+1)\lfloor lb(n-1) \rfloor + 4n + 2n + 3 > > 2(n+1)(lb(n-1)-1) + 4n + 2n + 3 = 2(n+1)lb(n-1) + 4n + 1,$$

$$l_2(n,n) < 2(n-n+1)(lbn+2) + 9n - 6 = 2lbn + 9n - 2.$$

Для нижней оценки $q(n)$ будем учитывать только часть описания последовательности, а именно номера вершин $D(a_i)$ и $D(d_i)$. Каждая вершина инцидентна двум рёбрам из последовательности всех рёбер.

$$\begin{aligned} \text{Поэтому } q(n) &> 2\sum\{D(i) \mid i=0..n-1\} = 6 + 2\sum\{2(\lfloor lbi \rfloor + 1) + 1 \mid i=1..n-1\} > \\ &> 2\sum\{2(lbi-1+1)+1 \mid i=1..n-1\} = 4\sum\{lbi \mid i=1..n-1\} + n(n-1) = 4lb((n-1)!) + n(n-1) > \\ &> 4lb((2\pi)^{1/2}(n-1)^{n-1+1/2}/e^{n-1}) + n(n-1) > 4(n-1)lb((n-1)/e) + n(n-1). \end{aligned}$$

Тогда $q(n) / l_1(n, n) > (4(n-1)lb((n-1)/e) + n(n-1)) / (2(n+1)lbn + 6n + 3)$ и

$$l_1(n, n) / l_2(n, n) > (2(n+1)lb(n-1) + 4n + 1) / (2lbn + 9n - 2).$$

При $n \rightarrow \infty$ правые части неравенств стремятся к бесконечности, т.е. $q(n)$ по порядку больше, чем $l_1(n, n)$, а $l_1(n, n)$ по порядку больше, чем $l_2(n, n)$.

Для графов с n вершинами без кратных рёбер и петель, поскольку $\Delta \leq n-1$, имеем $q(h) = O(n \log n \Delta) = O(n \log n^2) = O(n \log n) = o(n^2 \log n)$, т.е. меньше по порядку, чем размер памяти, необходимый для хранения графа.

6. Память для хранения корневого поддерева

Если нужно хранить не произвольное множество рёбер, а поддерево с корнем в корне графа, то требуется меньше памяти. При естественной ориентации дерева от корня к листьям ребро $a \rightarrow b$ задаётся его номером в вершине a .

Для представления такого дерева мы будем использовать естественную нумерацию вершин дерева, основанная на его обходе в ширину. Процедура нумерации является упрощённой версией процедуры нумерации корневого упорядоченного графа, описанная в подразделе 3.1. Корень получает номер 0. Выбирается первое ребро, инцидентное корню, и его конец получает номер 1. Далее выбирается следующее ребро, инцидентное корню, и его конец получает номер 2. И так далее. Когда исчерпаются все рёбра, инцидентные корню, аналогичная процедура выполняется для вершины 1. В общем, на

каждом шаге у нас определены номера вершин $0, \dots, a'$ и перебраны все рёбра дерева, инцидентные вершинам $0, \dots, a-1$, где $a < a'$, а для вершины a перебраны рёбра с номерами $1, \dots, i-1$. Если $i \leq \Delta(a)$, то для вершины a рассматривается ребро с номером i . Конец этого ребра получает номер $a'+1$. Когда все рёбра, инцидентные вершине с номером a , будут перебраны, т.е. когда $i > \Delta(a)$, переходим к следующей нумерованной вершине $a+1$, если $a < a'$. Если же $a = a'$, то процедура нумерации вершин закончена.

Располагаем описания вершин дерева в порядке этой нумерации, а для каждой вершины располагаем номера рёбер дерева, выходящих из этой вершины в естественной ориентации дерева от корня, в порядке возрастания номеров рёбер. Номер x выходящего ребра задаётся в формате $D(x)$. Оно занимает $2r(x)+1 \leq 2r(\Delta)+1 = 2(\lfloor \lg \Delta \rfloor + 1) + 1 < 2\lg \Delta + 3$ бит. Конец списка рёбер дерева, выходящих из данной вершины, задаётся кодом 10 или кодом 11 для последней вершины в списке вершин. Поскольку в дереве число рёбер не больше $n-1$, число вершин не больше n , а $\Delta \leq 2m$, такое представление дерева имеет размер не больше $(n-1)(2\lg \Delta + 3) + 2n = O(n \log \Delta) = O(n \log m) = o(n + (m-n+1) \log n)$ при любом (но фиксированном) n и $m \rightarrow \infty$.

Для графов с n вершинами без кратных рёбер и петель $\Delta \leq n-1$, поэтому представление дерева имеет размер $O(n \log \Delta) = O(n \log n) = o(n^2 \log n)$, т.е. меньше по порядку, чем размер памяти, необходимый для хранения графа.

7. Заключение

Мы исследовали размер памяти необходимый и достаточный для хранения неориентированного упорядоченного корневого графа. В разделе 2 изучался класс $G_1(n, m)$ нумерованных, а в разделе 3 – класс $G_2(n, m)$ нenumерованных графов с n вершинами и m рёбрами, в разделе 4 – классы $G_1(n)$ нумерованных

и $G_2(n)$ ненумерованных графов без кратных рёбер и петель с n вершинами.

Соответствующие оценки равны $\Theta(m \log n)$, $\Theta(n + (m - n + 1) \log n)$ и $\Theta(n^2 \log n)$.

Эти результаты мы предполагаем, в частности, использовать в последующих статьях, посвященных решению задач на неориентированных графах с помощью коллектива автоматов. Алгоритмы работы таких автоматов существенно зависят от размера памяти каждого автомата, определяемой числом состояний и числом входных и выходных символов. Если память автомата ограничена сверху константой, это конечный автомат (робот). Если память автомата зависит от размера графа, т.е. числа его вершин n и числа его рёбер m , то мы разделяем автоматы на неограниченные автоматы и полуроботы в зависимости от того, помещается в память автомата описание любого графа из рассматриваемого класса или нет. Тем самым, указанные выше оценки размера памяти определяют границу между неограниченным автоматом и полуроботом на соответствующих классах графов.

В разделе 5 показано, что размер памяти, достаточный для хранения последовательности $O(n)$ рёбер графа, равен $O(n \log n \Delta) = O(n \log n m)$ на классах $G_1(n, m)$ и $G_2(n, m)$, и $O(n \log n \Delta) = O(n \log n)$ на классах $G_1(n)$ и $G_2(n)$, что меньше по порядку, чем размер памяти, необходимый для хранения графа соответствующего класса. Тем самым, автомат, с памятью такого размера является полуроботом на соответствующих классах графов. В разделе 6 показано, что для хранения поддеревя с корнем в корне графа достаточно ещё меньше памяти: $O(n \log \Delta) = O(n \log m)$ на классах $G_1(n, m)$ и $G_2(n, m)$, и $O(n \log \Delta) = O(n \log n)$ на классах $G_1(n)$ и $G_2(n)$.

Список литературы

[1]. О.Оре. Теория графов. М., Наука, 1968.

[2]. F. Harary and E. M. Palmer, Graphical Enumeration, Academic Press, NY, 1973.

И. Бурдонов, А. Косачев.

Размер памяти для хранения упорядоченного корневого графа.

Труды Института системного программирования РАН. Том 29, выпуск 2, 2017, стр. 7-26. ISSN 2220-6426 (Online), ISSN 2079-8156 (Print).

20 стр.

[3]. <https://oeis.org/A000108>.

[4]. J. H. Conway and R. K. Guy, *The Book of Numbers*, New York: Springer-Verlag, 1995.

[5]. F. Bergeron, G. Labelle and P. Leroux, *Combinatorial Species and Tree-like Structures*, EMA vol.67, Cambridge, 1998.

[6]. R. Dutton, and R. Brigham, *Computationally Efficient. Bounds for the Catalan Numbers*. *Europ. J. Combinatorics*, vol.7, 1986.

[7]. <http://math24.biz/>.

Размер памяти для хранения упорядоченного корневого графа

Size of the memory for storage of ordered rooted graph

Igor Burdonov <igor@ispras.ru>

Alexander Kossatchev kos@ispras.ru

*Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*

Abstract. The paper considers boundaries of memory necessary and sufficient for storage of undirected ordered rooted connected graphs, both numbered and unnumbered. The introduction contains the basic definitions and the problem statement. A graph is rooted if one of its vertices is marked as a root. A graph is ordered if for each of its vertices all the incident edges are ordered (numbered). A graph is numbered if all its vertices are numbered with different integer numbers (from 0 to $n-1$, where n is the number of vertices). Two undirected ordered graphs G and G' are weakly isomorphic if there exists a one-to-one correspondence between their vertices, for which corresponding vertices has the same degrees (numbers of incident edges) and two edges having corresponding ends and the same numbers in these ends, also have the other ends corresponding. Isomorphism of rooted graphs should also correspond their roots. Isomorphism of numbered graphs should also correspond the vertices with the same numbers. Graphs are considered up to weak isomorphism. It is shown that the memory necessary and sufficient for storage of any graph has the size $\Theta(m \log n)$ for numbered graphs, $\Theta(n + (m-n+1) \log n)$ for unnumbered graphs with the number of vertices n and the number of edges m , and $\Theta(n^2 \log n)$ for graphs without multiple edges and loops with the number of vertices n . It is also shown that the memory sufficient for storage of an edge sequence of length $O(n)$ or a spanning tree, has the $O(n \log(n\Delta))$ or $O(n \log \Delta)$ size, respectively, where Δ is the maximum vertex degree.

Keywords: undirected graphs, ordered graph, labeled graph, rooted graph, graph representation, graph enumeration.

References

- [1]. O.Ore. Theory of graphs. AMS, Providence, Rhode Island, 1962.

И. Бурдонов, А. Косачев.

Размер памяти для хранения упорядоченного корневого графа.

Труды Института системного программирования РАН. Том 29, выпуск 2, 2017, стр. 7-26. ISSN 2220-6426 (Online), ISSN 2079-8156 (Print).

20 стр.

[2]. F. Harary and E. M. Palmer, Graphical Enumeration, Academic Press, NY, 1973.

[3]. <https://oeis.org/A000108>.

[4]. J. H. Conway and R. K. Guy, The Book of Numbers, New York: Springer-Verlag, 1995.

[5]. F. Bergeron, G. Labelle and P. Leroux, Combinatorial Species and Tree-like Structures, EMA vol.67, Cambridge, 1998.

[6]. R. Dutton, and R. Brigham, Computationally Efficient. Bounds for the Catalan Numbers. Europ. J. Combinatorics, vol.7, 1986.

[7]. <http://math24.biz/>.

Размер памяти для хранения упорядоченного корневого

графа

1. Введение

2. Нумерованные графы

2.1. Представление графа в памяти

**2.2. Хранение графов с меньшим числом вершин и/или
рёбер 10**

2.3. Число графов

3. Ненумерованные графы

3.1. Представление графа в памяти

**3.2. Хранение графов с меньшим числом вершин и/или
рёбер 17**

3.3. Число графов

4. Графы без кратных рёбер и петель

5. Память для хранения последовательности $O(n)$ рёбер

6. Память для хранения корневого поддерева

7. Заключение

Список литературы

References