

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 512.54

ОБ ОДНОМ АЛГОРИТМЕ ВЫЧИСЛЕНИЯ ФУНКЦИЙ РОСТА В КОНЕЧНЫХ ДВУПОРОЖДЁННЫХ ГРУППАХ ПЕРИОДА 5¹

А. А. Кузнецов

Сибирский государственный аэрокосмический университет, г. Красноярск, Россия

Пусть $B_0(2, 5) = \langle a_1, a_2 \rangle$ — наибольшая конечная двупорождённая бернсайдова группа периода 5, порядок которой равен 5^{34} . Для каждого элемента данной группы существует уникальное коммутаторное представление вида $a_1^{\alpha_1} \cdot a_2^{\alpha_2} \cdot \dots \cdot a_{34}^{\alpha_{34}}$, где $\alpha_i \in \mathbb{Z}_5$, $i = 1, 2, \dots, 34$. Здесь a_1 и a_2 — порождающие элементы $B_0(2, 5)$; a_3, \dots, a_{34} — коммутаторы, которые вычисляются рекурсивно через a_1 и a_2 . Определим фактор-группу группы $B_0(2, 5)$ следующего вида: $B_k = B_0(2, 5) / \langle a_{k+1}, \dots, a_{34} \rangle$. Очевидно, что $|B_k| = 5^k$. Предложен новый алгоритм, при помощи которого вычислены функции роста B_k относительно порождающих множеств $\{a_1, a_2\}$ и $\{a_1, a_1^{-1}, a_2, a_2^{-1}\}$ для $k = 15, 16, 17$. На основе полученных данных вычислены диаметр и средний диаметр соответствующих графов Кэли.

Ключевые слова: *функция роста, группа Бернсайда, граф Кэли.*

DOI 10.17223/20710410/33/10

AN ALGORITHM FOR COMPUTATION OF THE GROWTH FUNCTIONS IN FINITE TWO-GENERATED GROUPS OF EXPONENT 5

A. A. Kuznetsov

Siberian State Aerospace University, Krasnoyarsk, Russia

E-mail: alex_kuznetsov80@mail.ru

Let $B_0(2, 5) = \langle a_1, a_2 \rangle$ be the largest 2-generator Burnside group of exponent 5. It has the order 5^{34} . There is a power commutator representation of $B_0(2, 5)$. In this case, every element of the group can be uniquely represented as $a_1^{\alpha_1} \cdot a_2^{\alpha_2} \cdot \dots \cdot a_{34}^{\alpha_{34}}$, where $\alpha_i \in \mathbb{Z}_5$, $a_i \in B_0(2, 5)$, $i = 1, 2, \dots, 34$. Here, a_1 and a_2 are generators of $B_0(2, 5)$, commutators a_3, \dots, a_{34} are recursively defined by a_1 and a_2 . We define $B_k = B_0(2, 5) / \langle a_{k+1}, \dots, a_{34} \rangle$ as a quotient of $B_0(2, 5)$. It is clearly that $|B_k| = 5^k$. A new algorithm for computing the growth function of B_k is created. Using this algorithm, we calculated the growth functions of B_k relative to generating sets $\{a_1, a_2\}$ and $\{a_1, a_1^{-1}, a_2, a_2^{-1}\}$ for $k = 15, 16, 17$.

Keywords: *Burnside group, the growth function.*

¹Работа поддержана грантом Президента РФ (проект № МД-3952.2015.9).

Введение

Одним из важных инструментов для определения строения группы является изучение её роста относительно фиксированного порождающего множества. Пусть $G = \langle X \rangle$. Шаром K_s радиуса s группы G будем называть множество всех её элементов, которые могут быть представлены в виде групповых слов в алфавите X длины не больше s . Для каждого целого неотрицательного s можно определить функцию $F(s)$ роста группы, которая равна числу элементов группы G относительно X , представимых в виде несократимых групповых слов длины s . Таким образом,

$$F(0) = |K_0| = 1, \quad F(s) = |K_s| - |K_{s-1}| \text{ при } s \in \mathbb{N}.$$

Как правило, функцию роста конечной группы представляют в виде таблицы, в которую записывают ненулевые значения $F(s)$.

К сожалению, вычисление функции роста большой конечной группы является хотя и разрешимой, но весьма сложной проблемой. Это связано с тем, что в общем случае задача по определению минимального слова элемента группы, как показали С. Ивен и О. Голдрейх [1], является NP-трудной. Таким образом, в наихудшем случае количество элементарных операций, которые необходимо выполнить для решения указанной задачи, представляет собой экспоненциальную функцию от $|X|$.

Отметим также, что при вычислении функции роста группы мы параллельно выясняем характеристики соответствующего графа Кэли, например диаметр и средний диаметр [2]. Пусть $F(s_0) > 0$, но $F(s_0 + 1) = 0$, тогда s_0 является диаметром графа Кэли группы G в алфавите порождающих X , который будем обозначать $D_X(G)$. Соответственно средний диаметр $\bar{D}_X(G)$ равен $\frac{1}{|G|} \sum_{s=0}^{s_0} s \cdot F(s)$.

Как известно, благодаря своим замечательным свойствам графы Кэли нашли применение при моделировании топологий многопроцессорных вычислительных систем (МВС) — суперкомпьютеров [3], а также дата-центров [4]. И вполне возможно, что в недалёком будущем понадобятся знания о сверхбольших графах, которые будут востребованы при проектировании распределённых систем, пиковая производительность которых будет достигать 1 эксафлопс и выше.

Одной из широко применяемых топологий МВС является k -мерный гиперкуб, который задаётся группой $B(k, 2)$ — k -порождённой бернсайдовой группой периода 2. Группа $B(k, 2)$ имеет простую структуру и равна прямому произведению k экземпляров циклической группы порядка 2. Обобщением гиперкуба является n -мерный тор, который порождается прямым произведением n экземпляров циклических подгрупп, порядка которых могут не совпадать. В работе [5] исследованы графы Кэли групп $B(k, 3)$, т. е. групп периода 3, а также проведён сравнительный анализ данных графов с гиперкубами. Анализ выявил, что графы $B(k, 3)$ обладают более предпочтительными характеристиками, чем $B(k, 2)$. Это означает, что при парном сравнении графов $B(k_1, 3)$ и $B(k_2, 2)$ с примерно одинаковым количеством вершин первые будут иметь меньшие диаметры, средние диаметры и степени. В связи с этим представляет интерес задача по исследованию графов Кэли конечных бернсайдовых групп другого периода.

Пусть $B_0(2, 5) = \langle a_1, a_2 \rangle$ — максимальная конечная двупорождённая бернсайдова группа периода 5, порядок которой равен 5^{34} [6]. Используя систему компьютерной алгебры GAP, нетрудно получить pc -представление (*Power Commutator presentation*) данной группы [3, 7]. В этом случае каждый элемент $g \in B_0(2, 5)$ может быть однозначно записан в следующем виде:

$$g = a_1^{\alpha_1} \cdot a_2^{\alpha_2} \cdot \dots \cdot a_{34}^{\alpha_{34}}, \quad \alpha_i \in \mathbb{Z}_5, \quad i = 1, 2, \dots, 34.$$

Здесь a_1 и a_2 — порождающие элементы $B_0(2, 5)$; a_3, \dots, a_{34} — коммутаторы, которые вычисляются рекурсивно через a_1 и a_2 [6].

Обозначим через B_k фактор-группу $B_0(2, 5)$ следующего вида:

$$B_k = B_0(2, 5) / \langle a_{k+1}, \dots, a_{34} \rangle.$$

Очевидно, что $|B_k| = 5^k$ и для всех $g \in B_k$ выполняется

$$g = a_1^{\alpha_1} \cdot a_2^{\alpha_2} \cdot \dots \cdot a_k^{\alpha_k}.$$

Помимо прикладного интереса, существует и другая причина повышенного внимания исследователей к функциям роста B_k и особенно к $B_{34} = B_0(2, 5)$. Это объясняется тем, что полученная информация может оказаться полезной при решении открытой проблемы о конечности $B(2, 5)$ — свободной дупорождённой бернсайдовой группы периода 5. Если $B(2, 5)$ конечна, то $B_{34} = B(2, 5)$. Однако в обозримом будущем вычислить функцию роста B_{34} едва ли возможно, поскольку количество её элементов очень велико: $5^{34} = 582076609134674072265625 \approx 5 \cdot 10^{23}$.

К. А. Филипповым в работе [8] вычислена функция роста группы B_{14} в алфавите $A_2 = \{a_1, a_2\}$. Аналогичная задача для симметричного порождающего множества $A_4 = \{a_1, a_1^{-1}, a_2, a_2^{-1}\}$ решена Ч. Симсом [9]. В обоих случаях решение было получено при помощи длительных компьютерных вычислений. Отметим также, что Симс в своём алгоритме использовал элегантный трюк, основанный на применении к B_{14} группы автоморфизмов, сохраняющих длину группового слова. Это дало возможность свести задачу к рассмотрению только 1/16 части элементов группы. Подобный трюк для несимметричного порождающего множества менее эффективен, поскольку позволяет сократить количество элементов не более чем в 2 раза.

Дальнейшее изучение функций роста групп B_k для $k > 14$ сталкивается с серьёзными вычислительными трудностями ввиду их больших порядков.

В данной работе представлен эффективный алгоритм для вычисления функций роста указанных групп, который даёт возможность преодолеть существующий барьер. Компьютерная реализация алгоритма позволила вычислить функции роста групп B_k для $k = 15, 16$ и 17 .

В п. 1 рассматривается базовый алгоритм для вычисления функции роста конечной группы, а также исследуется его сложность. В п. 2 алгоритм модифицируется для исследования групп B_k . В п. 3 приведены результаты компьютерных вычислений по модифицированному алгоритму.

1. Описание базового алгоритма

В качестве отправной точки далее представлен общий алгоритм А-I, вычисляющий функцию роста произвольной конечной группы G , заданной фиксированным порождающим множеством X .

Лемма 1. Алгоритм А-I корректен, т. е. он за конечное число шагов вычислит функцию роста любой конечной группы G , заданной фиксированным порождающим множеством X .

Доказательство. По построению А-I выражает каждый элемент группы G в виде группового слова наименьшей длины в алфавите X . После каждого прохода от п. 2 до п. 8 множество K_s представляет собой шар радиуса s группы G относительно X . Поскольку количество элементов группы конечно, через конечное число шагов, при некотором $s = s_0 + 1$, все $g \in G$ будут записаны в виде слов, длины которых меньше

Алгоритм 1. А-I**Вход:** $X = \{x_1, x_2, \dots, x_m\}$ — порождающее множество G **Выход:** функция роста $F(s)$ группы G 1: $s := 0, F(0) := 1, K_0 := \{e\}, P := K_0$ 2: $s := s + 1$ 3: $K_s := K_{s-1}$ 4: **Для всех** $x \in X$ и $p \in P$ 5: $g = x \cdot p$ (или $g = p \cdot x$)6: **Если** $g \notin K_s$, **то**
 $K_s := K_s \cup \{g\}$ 7: $P := K_s - K_{s-1}$ 8: $F(s) := |P|$ 9: **Если** $F(s) > 0$, **то**
переход в п. 2,10: **иначе**11: $s_0 := s - 1$, переход в п. 1212: $D_X(G) := s_0, \bar{D}_X(G) := \frac{1}{|K_{s_0}|} \sum_{s=0}^{s_0} s \cdot F(s)$, **выход**

$s_0 + 1$. Получим $F(s_0 + 1) = 0$ и $|K_{s_0}| = |G|$, что означает остановку алгоритма. При этом $D_X(G) = s_0$ и $\bar{D}_X(G) = \frac{1}{|K_{s_0}|} \sum_{s=0}^{s_0} s \cdot F(s)$. ■

Несмотря на то, что алгоритм А-I хорошо известен специалистам, автору не удалось найти публикаций, в которых были бы вычислены оценки его временной сложности. Например, в системе компьютерной алгебры GAP реализован указанный алгоритм, вычисляющий функцию роста группы подстановок. Однако в документации к системе отсутствуют сведения о его сложности. Для устранения указанного пробела воспользуемся машиной с произвольным доступом к памяти — RAM-машиной, а также асимптотическим анализом сложности [10]. Далее термины «временная сложность» и «сложность» будем считать равнозначными; кроме этого, введём обозначения:

- $T(|G|)$ — сложность алгоритма А-I;
- $O(f(|G|))$ — верхняя асимптотическая оценка сложности;
- $\Omega(f(|G|))$ — нижняя асимптотическая оценка сложности;
- $\Theta(f(|G|))$ — одновременно верхняя и нижняя оценки сложности.

Нам понадобятся следующие вспомогательные утверждения.

Лемма 2. В процессе работы алгоритма А-I необходимо вычислить и проверить $|X| \cdot |G|$ групповых слов.

Доказательство. При вычислении слов длины $s \geq 1$ следует обработать $F(s - 1)|X|$ элементов. В итоге получим

$$\sum_{s=1}^{D_X(G)+1} F(s-1)|X| = |X| \sum_{s=1}^{D_X(G)+1} F(s-1) = |X| \cdot |G|.$$

Лемма доказана. ■

Лемма 3. Пусть $N(|G|)$ — суммарное количество проверок вида $g \notin K_s$ алгоритма А-I. Тогда верно следующее неравенство:

$$\frac{1}{2}|G|^2 + \left(|X| - \frac{3}{2}\right)|G| + 1 \leq N(|G|) \leq \left(|X| - \frac{1}{2}\right)|G|^2 + \frac{1}{2}|G|.$$

Доказательство. Рассмотрим сначала те групповые слова, которые войдут в K_s . Всего таких элементов, не считая единицы, $(|G| - 1)$. Для первого из них необходимо осуществить одну проверку (с единицей группы), для второго — две и т. д. В итоге получим $1 + 2 + \dots + (|G| - 1) = |G|(|G| - 1)/2$ операций.

Теперь рассмотрим те слова, которые не войдут в K_s . Согласно лемме 2, количество таких элементов равно $|X| \cdot |G| - (|G| - 1)$. В лучшем случае каждый из этих элементов будет проверен один раз и исключён, а в худшем — для каждого из них потребуется осуществить $|G|$ сравнений.

Таким образом, общее число проверок в худшем и лучшем случаях равно $|G|(|G| - 1)/2 + |X| \cdot |G| - (|G| - 1)$ и $|G|(|G| - 1)/2 + (|X| \cdot |G| - (|G| - 1))|G|$ соответственно. После элементарных преобразований получим нужное неравенство. ■

Теорема 1. $T(|G|) \in \Omega(|G|^2)$ и $T(|G|) \in O(|X| \cdot |G|^2)$.

Доказательство. При $|G| \rightarrow \infty$ сложностью операций всех пунктов алгоритма А-I, кроме четвертого, следует пренебречь. Поскольку количество элементарных операций, необходимое для вычисления произведения двух элементов группы, конечно и ограничено $\Theta(1)$, то суммарная сложность $M(|G|)$ п. 5 ограничена $\Theta(|X| \cdot |G|)$. Согласно лемме 3, получим $|G|^2/2 \leq N(|G|) \Rightarrow N(|G|) \in \Omega(|G|^2)$ и $N(|G|) \leq |X| \cdot |G|^2 \Rightarrow N(|G|) \in O(|X| \cdot |G|^2)$. Так как $T(|G|) = M(|G|) + N(|G|)$, получим $T(|G|) \in \Omega(|G|^2)$ и $T(|G|) \in O(|X| \cdot |G|^2)$. ■

Следствие 1. Если $|X| \ll |G|$, то $T(|G|) \in \Theta(|G|^2)$.

Следствие 2. Если $|X| \sim |G|$, то $T(|G|) \in O(|G|^3)$.

Если порядок группы G достаточно большой, то наиболее критическими участками алгоритма А-I будут пп. 5 и 6, т. е. умножение элементов группы и проверка, позволяющая определить, встречался ранее такой элемент или нет.

2. Модификация алгоритма А-I для исследования групп B_k

В данном случае $X = A_2$ или A_4 . Рассмотрим п. 5 алгоритма.

Существуют два способа умножения элементов в группах B_k : собирательный процесс [3, 7] и метод полиномов Холла [11]. Вычислительные эксперименты показали, что второй способ позволяет значительно быстрее собирательного процесса (по крайней мере на порядок) умножать элементы в группах B_k [12].

Следующий вопрос: как эффективнее умножать порождающие $x \in X$ на элементы группы $p \in B_k$ — слева (1 способ) или справа (2 способ), т. е. $x \cdot p$ или $p \cdot x$? Кроме этого, была предложена идея при умножении слева использовать пять вариантов полиномов в зависимости от значения степени α_1 элемента p (3 способ). Для ответа на данный вопрос необходимо вычислить суммарное число операций $t(k)$ в полиномах Холла, необходимое для выполнения одного произведения. В полиномах имеются три вида действий: сложение, умножение и остаток от деления числа на 5. На рис. 1 представлены графики $t(k)$ для каждого из трёх способов (в предположении, что указанные действия имеют одинаковую сложность равную 1).

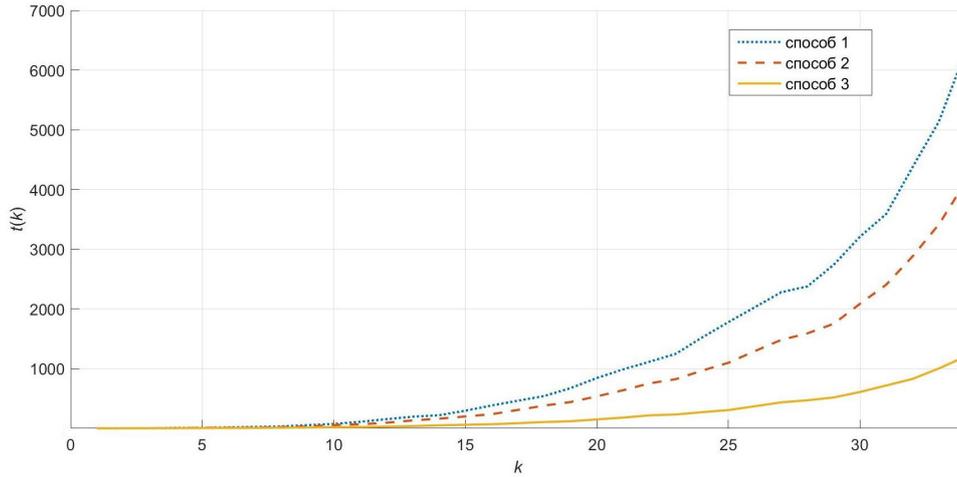
Рис. 1. Графики сложности $t(k)$ для одного произведения

Рис. 1 наглядно показывает, что третий способ самый эффективный. Этот факт был учтён при реализации п. 5 алгоритма.

Теперь обратимся к п. 6 алгоритма А-І. Для его оптимизации потребуется каждому элементу $g \in B_k$ присвоить уникальный порядковый номер. Пусть $g = a_1^{\alpha_1} \cdot a_2^{\alpha_2} \cdot \dots \cdot a_k^{\alpha_k}$ — произвольный элемент B_k . Определим биективное отображение ϕ следующего вида:

$$g \xleftrightarrow{\phi} n_g = \alpha_1 \alpha_2 \dots \alpha_k.$$

Здесь n_g представляет собой целое неотрицательное число, записанное в пятеричной системе счисления, которое возьмём в качестве порядкового номера g . Легко увидеть, что n_g пробегает все значения от 0 до $(5^k - 1)$.

Модифицируем А-І следующим образом. В п. 1 добавим булев вектор V размерности 5^k , все элементы которого первоначально равны 0. Для удобства индексация элементов V начинается с 0, т. е. последний элемент вектора имеет индекс $(5^k - 1)$. Заменяем п. 6 алгоритма А-І на следующий:

6: **Если** $V_{n_g} = 0$, **то** $V_{n_g} := 1$ и $K_s := K_s \cup \{g\}$.

Данная модификация позволяет избежать трудоёмкой процедуры поиска элемента g в множестве K_s . Будем считать, что для выполнения п. 6 алгоритма требуется не более $t(g)$ элементарных операций.

Несмотря на то, что порядки B_k ограничены сверху, имеет смысл рассмотреть асимптотическую оценку сложности модифицированной версии алгоритма, поскольку данные группы содержат достаточно большое число элементов при $k > 8$.

Теорема 2. Пусть $T(|B_k|)$ — вычислительная сложность модифицированного алгоритма А-І. Тогда верно, что $T(|B_k|) \in \Theta(|B_k|)$.

Доказательство. При вычислении асимптотической оценки следует учитывать сложность только наиболее трудоёмких фрагментов алгоритма, т. е. пп. 5 и 6. Поскольку $T(|G|) = (t(k) + t(g))|X| \cdot |B_k|$, $|X| \ll |B_k|$, $t(k) \ll |B_k|$ и $t(g) \ll |B_k|$, получим $T(G) \in \Theta(|B_k|)$. ■

Также отметим, что в п. 5 все элементы g вычисляются независимо друг от друга, поэтому этот участок алгоритма можно легко распараллелить. В этом случае сначала параллельно вычисляются все произведения g , а затем для каждого элемента получившегося массива последовательно выполняется п. 6.

Легко увидеть, что перечисленные модификации алгоритма А-І применительно к группам B_k значительно повышают его производительность.

3. Компьютерная реализация алгоритма

Модифицированный алгоритм А-І реализован на языке С++ и апробирован на 4-процессорном компьютере (на каждом по 16 ядер) с общей памятью объёмом 256 Гбайт. В результате вычислены функции роста групп B_k для $k = 15, 16$ и 17 для порождающих множеств $A_2 = \{a_1, a_2\}$ и $A_4 = \{a_1, a_1^{-1}, a_2, a_2^{-1}\}$. Затраченное на расчёты время примерно равно 1 час для $k = 15$, 5 часов — для $k = 16$ и около суток — для $k = 17$. На рис. 2–7 представлены графики функций роста указанных групп.

В таблице приведены характеристики графов Кэли групп B_k , вычисленные в настоящей работе, а также, для полноты картины, уже известные. Значения средних диаметров округлены до ближайших целых.

Отметим, что для решения задачи маршрутизации в графе Кэли, после того как будут получены минимальные слова группы в результате работы алгоритма А-І, можно будет воспользоваться алгоритмом из [4], который имеет сложность $O(D_X^2(G))$.

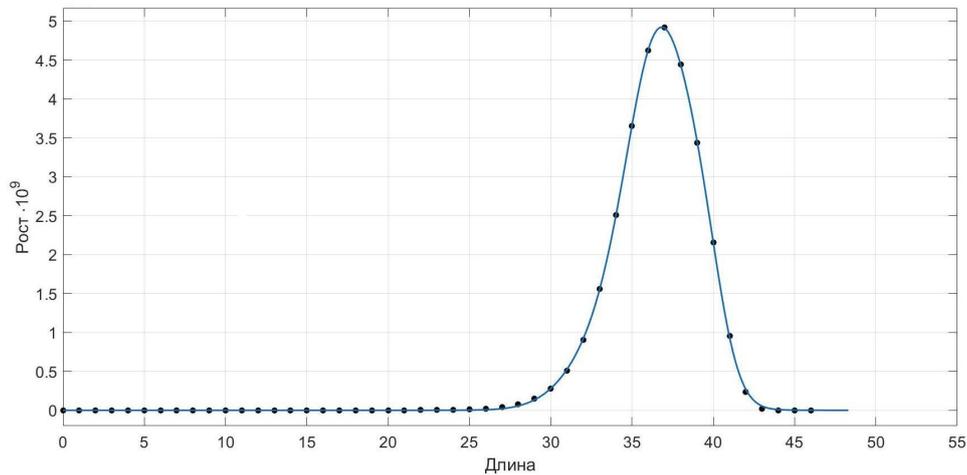


Рис. 2. График функции роста группы B_{15} в алфавите A_2

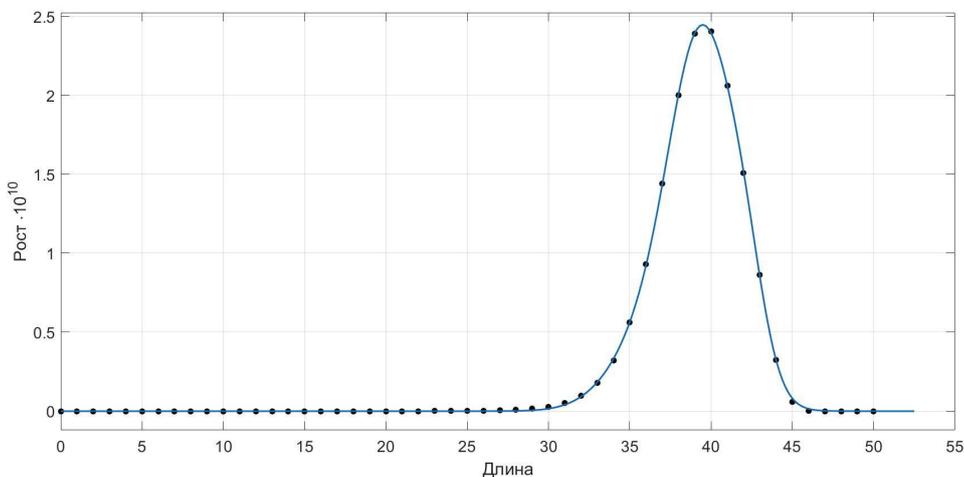


Рис. 3. График функции роста группы B_{16} в алфавите A_2

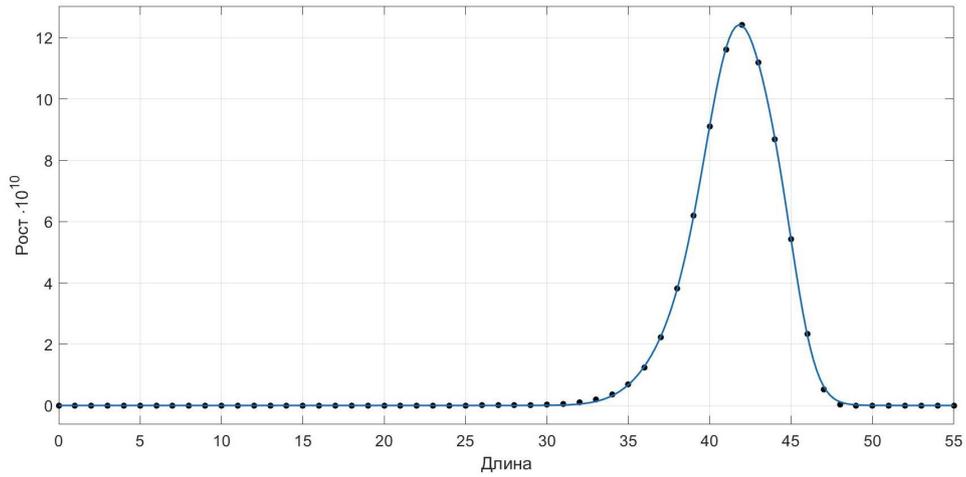


Рис. 4. График функции роста группы B_{17} в алфавите A_2

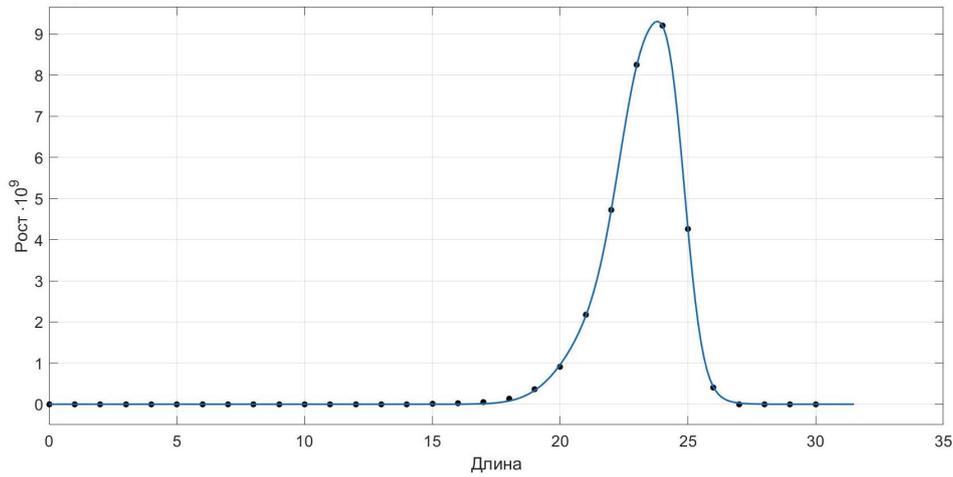


Рис. 5. График функции роста группы B_{15} в алфавите A_4

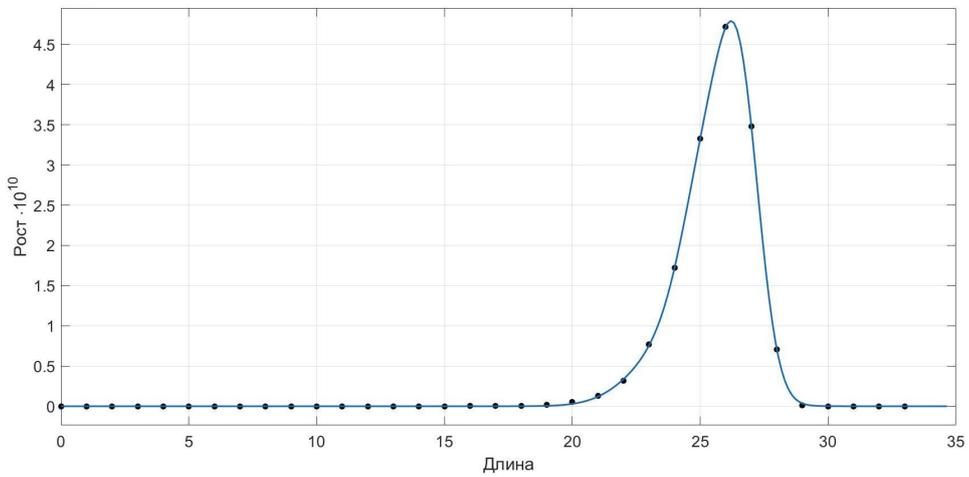
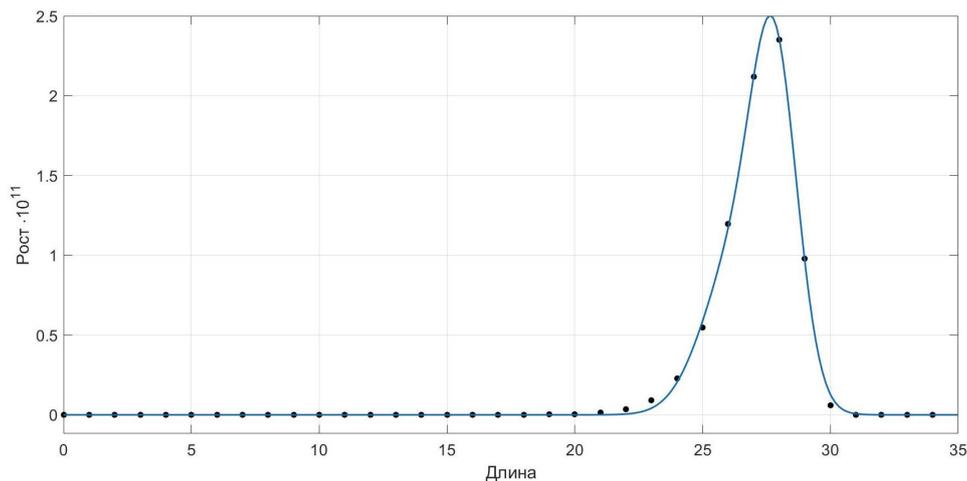


Рис. 6. График функции роста группы B_{16} в алфавите A_4

Рис. 7. График функции роста группы B_{17} в алфавите A_4 Характеристики графов Кэли групп B_k

k	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$D_{A_2}(B_k)$	8	10	13	20	20	25	30	31	32	35	40	40	45	46	50	55
$\overline{D}_{A_2}(B_k)$	4	6	8	12	14	16	20	22	24	27	29	31	34	36	39	42
$D_{A_4}(B_k)$	4	6	9	10	13	15	19	20	20	24	26	28	30	30	33	34
$\overline{D}_{A_4}(B_k)$	2	4	5	7	9	10	13	14	15	17	19	21	22	23	26	27

ЛИТЕРАТУРА

1. *Even S. and Goldreich O.* The Minimum Length Generator Sequence is NP-hard // J. Algorithms. 1981. No. 2. P. 311–313.
2. *Кузнецов А. А., Кузнецова А. С.* Параллельный алгоритм для исследования графов Кэли групп подстановок // Вестник СибГАУ. 2014. № 1. С. 34–39.
3. *Holt D., Eick B., and O'Brien E.* Handbook of Computational Group Theory. Boca Raton: Chapman & Hall/CRC Press, 2005. 514 p.
4. *Camelo M., Papadimitriou D., Fàbrega L., and Vilà P.* Efficient routing in data center with underlying Cayley graph // Proc. 5th Workshop on Complex Networks CompleNet. 2014. P. 189–197.
5. *Кузнецов А. А.* Графы Кэли бернсайдовых групп периода 3 // Сибирские электронные математические известия. 2015. Т. 12. С. 248–254.
6. *Havas G., Wall G., and Wamsley J.* The two generator restricted Burnside group of exponent five // Bull. Austral. Math. Soc. 1974. No. 10. P. 459–470.
7. *Sims C.* Fast multiplication and growth in groups // Proc. 1998 Intern. Symp. Symbolic Algebraic Computation. 1998. P. 165–170.
8. *Филлипов К. А.* О диаметре Кэли одной подгруппы группы $B_0(2, 5)$ // Вестник СибГАУ. 2012. № 1. С. 234–236.
9. *Sims C.* Computation with Finitely Presented Groups. Cambridge: Cambridge University Press, 1994. 628 p.
10. *Скиена С.* Алгоритмы. Руководство по разработке. СПб.: БХВ-Петербург, 2014. 720 с.
11. *Hall P.* Nilpotent Groups: Notes of Lectures Given at the Canadian Mathematical Congress Summer Seminar, University of Alberta, 12–30 August, 1957. London: Queen Mary College, 1969.

12. Кузнецов А. А., Кузнецова А. С. Быстрое умножение элементов в конечных двупорождённых группах периода пять // Прикладная дискретная математика. 2013. № 1. С. 110–116.

REFERENCES

1. Even S. and Goldreich O. The Minimum Length Generator Sequence is NP-hard. J. Algorithms, 1981, no. 2, pp. 311–313.
2. Kuznetsov A. A. and Kuznetsova A. S. Parallel'nyy algoritm dlya issledovaniya grafov Keli grupp podstanovok [A parallel algorithm for study of the Cayley graphs of permutation groups]. Vestnik SibGAU, 2014, no. 1, pp. 34–39. (in Russian)
3. Holt D., Eick B., and O'Brien E. Handbook of Computational Group Theory. Boca Raton, Chapman & Hall/CRC Press, 2005. 514 p.
4. Camelo M., Papadimitriou D., Fàbrega L., and Vilà P. Efficient routing in data center with underlying Cayley graph. Proc. 5th Workshop on Complex Networks CompleNet, 2014, pp. 189–197.
5. Kuznetsov A. A. Grafy Keli bernsайдovykh grupp perioda 3 [The Cayley graphs of Burnside groups of exponent 3]. Siberian Electronic Math. Reports, 2015, vol. 12, pp. 248–254. (in Russian)
6. Havas G., Wall G., and Wamsley J. The two generator restricted Burnside group of exponent five. Bull. Austral. Math. Soc., 1974, no. 10, pp. 459–470.
7. Sims C. Fast multiplication and growth in groups. Proc. 1998 Intern. Symp. Symbolic Algebraic Computation, 1998, pp. 165–170.
8. Filippov K. A. O diametre Keli odnoy podgruppy gruppy $B_0(2, 5)$ [About Cayley diameter of one subgroup of the group $B_0(2, 5)$]. Vestnik SibGAU, 2012, no. 1, pp. 234–236. (in Russian)
9. Sims C. Computation with Finitely Presented Groups. Cambridge, Cambridge University Press, 1994. 628 p.
10. Skiena C. Algoritmy. Rukovodstvo po razrabotke [Algorithms. Guide for the Development]. SPb., BKhV-Petersburg Publ., 2014. (in Russian)
11. Hall P. Nilpotent Groups: Notes of Lectures Given at the Canadian Mathematical Congress Summer Seminar, University of Alberta, 12–30 August, 1957. London, Queen Mary College, 1969.
12. Kuznetsov A. A. and Kuznetsova A. S. Bystroe umnozhenie elementov v konechnykh dvuporozhdennykh gruppakh perioda pyat' [Fast multiplication in finite two-generated groups of exponent five]. Prikladnaya Diskretnaya Matematika, 2013, no. 1, pp. 110–116. (in Russian)