Linear Space Data Structures for Finite Groups with Constant Query-Time

Bireswar Das[1]
Indian Institute of Technology Gandhinagar, India

Anant Kumar
Indian Institute of Technology Gandhinagar, India

Shivdutt Sharma
Indian Institute of Information Technology, Una, India

Dhara Thakkar
Indian Institute of Technology Gandhinagar, India

Abstract

A finite group of order n can be represented by its Cayley table. In the word-RAM model the Cayley table of a group of order n can be stored using $O(n^2)$ words and can be used to answer a multiplication query in constant time. It is interesting to ask if we can design a data structure to store a group of order n that uses $o(n^2)$ space but can still answer a multiplication query in constant time.

We design a constant query-time data structure that can store any finite group using $O(n)$ words where n is the order of the group.

Farzan and Munro (ISSAC 2006) gave an information theoretic lower bound of $\Omega(n)$ on the number of words to store a group of order n. Since our data structure achieves this lower bound and answers queries in constant time, it is optimal in both space usage and query-time.

A crucial step in the process is essentially to design linear space and constant query-time data structures for nonabelian simple groups. The data structures for nonableian simple groups are designed using a lemma that we prove using the Classification Theorem for Finite Simple Groups (CFSG).

2012 ACM Subject Classification Theory of computation → Data structures design and analysis; Theory of computation → Data compression

Keywords and phrases Compact Data Structures, Space Efficient Representations, Finite Groups, Simple Groups, Classification Theorem for Finite Simple Groups

Digital Object Identifier 10.4230/LIPIcs.STACS.2022.25

Funding *Bireswar Das*: Funded by Ministry of Education, Govt. of India.
Dhara Thakkar: Funded by CSIR-UGC NET JRF Fellowship.

1 Introduction

The Cayley table of a group of order n is a two dimensional table whose (i, j)th entry is the product of the ith and jth element of the group. In the word-RAM model while it takes $O(n^2)$ words to store the Cayley table of a group of order n, a multiplication query can be answered in constant time by accessing the appropriate location of the table.

For many computational problems in group theory the input group is given by its Cayley table. Some of these problems include the minimum generating set problem, various problems in property testing, the group factoring problem, and the group isomorphism

[1] Corresponding author.

problem [18, 1, 15, 19]. Among these, the group isomorphism problem is probably the most prominent one because of its unresolved complexity status despite years of extensive research [4, 13, 2, 5, 20, 14].

The Cayley table is very fast in terms of query processing but it takes quadratic space to store a group. It is interesting to ask if we can design a data structure for finite groups using $o(n^2)$ space[2] which can still answer multiplication query in constant time. We note that while quasigroups, and semigroups can also be stored using their Cayley tables, it is not possible to store quasigroups, and semigroups using $o(n^2)$ space. This is simply because the numbers of quasigroups, and semigroups are too large [26, 17] and the information theoretic lower bound is $\Omega(n^2 \log n)$ bits or $\Omega(n^2)$ words.

Das et al. [9] showed that for any finite group G of order n and for any $\delta \in [1/\log n, 1]$, a data structure can be constructed for G that uses $O(n^{1+\delta}/\delta)$ space and answers a multiplication query in time $O(1/\delta)$. Their result implies that there exist constant query-time data structures for finite groups of order n that use $O(n^{1.01})$ space. However, the result cannot be used to design a constant query-time data structure even if we are allowed to use $\Theta(n.polylog(n))$ space.

In this paper we design constant query-time data structures for finite groups that can be stored using $O(n)$ words where n is the order of the group. An information theoretic argument by Farzan and Munro shows that a lower bound to store a group of order n is $\Omega(n \log n)$ bits or $\Omega(n)$ words [11]. Our data structure is optimal in the sense that it achieves the lower bound. A data structure that achieves the optimum information theoretic lower bound asymptotically is known as *a compact data structure*. Therefore our data structure is a constant query-time compact data structure for finite groups. We note that compact query-time data structures were designed for some restricted classes of groups such as abelian groups and Dedekind groups [8].

In the process of designing the data structure we first prove two results, which we call *extension theorems*, on the construction of data structures for a group when we already have a data structure for a subgroup of the given group. The extra space used by the newly constructed data structure depends on the index of the subgroup in one of the results and the structure[3] of the subgroup in the other result. This indicates that finding suitable subgroups of a group might be useful.

The Jordan-Hölder theorem provides us with a supply of subgroups in the form of composition series. In our process we try to pick some suitable subgroups that are elements of the composition series of the given group. However, picking suitable groups is not always possible. This happens, as we will see in Section 4, when there is a "large" composition factor sitting in a certain position of the composition series. The composition factors are simple groups. In a sense the hard cases for constructing the data structure are for the simple groups.

Simple groups are sometimes considered as the building blocks for finite groups. The Classification Theorem for Finite Simple Groups (CFSG) is one of the most important theorems in group theory. Informally, this theorem classifies the finite simple groups into cyclic groups, alternating groups, certain groups of Lie-type and into 26 sporadic simple groups. The precise statement of the theorem could be found in Section 5. Except for the 26 sporadic simple groups the other group classes are infinite. We use CFSG to prove a key lemma that allows us to handle the case for the nonabelian simple groups.

---

[2] In this paper we use the word-RAM model. The space used by a data structure or an algorithm refers to the number of words used by them.

[3] The subgroup needs to be normal and quotient needs to be cyclic.

We note that for solvable groups the design of the data structure is *independent* of CFSG. The composition factors of a solvable group are cyclic of prime order. Such cases are handled using one of the extension theorems proved in Section 3.

Related work. Farzan and Munro [11] gave a succinct representations for finite abelian groups in a specific model of computation. In their model *a compression algorithm* first produces labels of each group element. The queries are processed by a *query processing unit* which is similar to the word-RAM model. However, along with the common arithmetic, logical and comparison operations the query processing unit can also perform bit-reversal in constant time. A user issuing a query, supplies the labels of two group elements that were generated by the compression algorithm to the query processing unit which then returns the label of the product of the two elements.

Das et al. [9] and Das and Sharma [8] have used Erdös-Réyni cube generating sequences, Remak-Krull-Schmidt decomposition and the structure of indecomposable groups to design their space and query-time efficient data structures. Our approach is quite different in the sense that we use the extension theorems (Section 3) and the Classification Theorem for Finite Simple groups to design the data structures.

Remark. There are several ways to represent a finite group apart from the Cayley table representation. The permutation group representation, the polycyclic presentations and the generator-relator presentations are some of the common group representations. These representations are often incomparable. For example in the generator-relator presentation we can represent infinite groups. However, many problems such as the membership testing, testing if a group is finite becomes undecidable in the generator-relator presentation (c.f. [25]). In the permutation group representation the membership testing takes superlinear time in terms of the degree of the representation and polylogarithmic in the order of the group [24, 23, 12]. We contrast this with the Cayley table representation where membership testing can be done in constant time since the elements are known and are already used as row and column indices of the Cayley table. In the Cayley representation the user knows the labels or the names of each group element explicitly and has a direct access to each element. The labels of the elements are often taken to be $1, 2, \ldots, n$ where n is the order of the group. The situation is quite different for permutation group representation, polycyclic presentation or generator-relator presentation. In these cases the user does not have an explicit representation for each element.

2 Preliminary

In this section we recall some definitions and notations which we use in this paper. In this paper we only consider finite groups. The number of elements in a group G is called the *order of* G and is denoted by $|G|$. A group G is abelian if $g_1g_2 = g_2g_1$ for all $g_1, g_2 \in G$. For a subgroup H of G and $g \in G$, the set $gH = \{gh \mid h \in H\}$ is called a left coset. Similarly, we can define right coset of G. The number of the left (or right) cosets of H in G is called the *index of* H *in* G and is denoted by $[G : H]$. A *left traversal* of H in G is a set containing exactly one element from each left coset and similarly we can define right traversals. The size of left (right) traversal is the same as the index $[G : H]$. For $g \in G$, the set $gHg^{-1} = \{gag^{-1} \mid a \in H\}$ is called a conjugate of the subgroup H. A subgroup H of G is said to be *normal* in G (denoted $H \trianglelefteq G$) if $gHg^{-1} = H$ for all $g \in G$. We define the *normalizer* of H in G to be the set $N_G(H) = \{g \in G \mid gHg^{-1} = H\}$. Note that, $N_G(H)$ is the largest subgroup in G in which H is normal.

A group G is called *simple* if G has no nontrivial normal subgroup. The Classification Theorem of Finite Simple Groups states that all the finite simple groups can be classified into the following five classes: (1) cyclic groups of prime order, (2) alternating groups, (3) classical groups, (4) exceptional groups of Lie type and (5) 26 sporadic simple groups.

We list all the classes of the finite simple groups later in the Classification Theorem for Simple Groups in Section 5. If G is a finite simple group of Lie-type over $\mathbb{F}_q$ where q is a power of some prime p, the Borel subgroup B of G is defined as the semidirect product of the Sylow p-subgroup of G with the maximal split torus T. The Borel subgroup is also the normalizer of the Sylow p-subgroup of the finite simple group (see [6], [27]).

For the purpose of this paper it might be sufficient to know some results on the *orders* of certain subgroups of simple groups. The reader may choose to skip the details of the structure of these groups. We indicate what kind of subgroups we are interested in and the results regarding the order of those subgroups as and when required. An interested reader may refer to the books by Carter [6], Wilson [27], or Aschbacher [3] for more details.

▶ **Definition 1** (see e.g., [10]). *A* subnormal series *of a group G is chain of subgroups*

$$1 = G_k \leq G_{k-1} \leq \cdots \leq G_1 \leq G_0 = G$$

such that $G_i \trianglelefteq G_{i-1}$, for all i.

▶ **Definition 2** (see e.g., [10]). *In a group G a sequence of subgroups*

$$1 = G_k \leq G_{k-1} \leq \cdots \leq G_1 \leq G_0 = G$$

is called a composition series *if $G_i \trianglelefteq G_{i-1}$ and G_{i-1}/G_i is simple for all $i \in [k]$. Here, k is the* composition length *of G.*

▶ **Theorem 3** (Jordan-Hölder Theorem see e.g., [10]). *Let G be a finite group with $G \neq 1$. Then*

- **(i)** *G has a composition series.*
- **(ii)** *The composition factors in a composition series are unique, namely, if $1 = N_r \leq N_{r-1} \leq \cdots \leq N_1 \leq N_0 = G$ and $1 = M_s \leq M_{s-1} \leq \cdots \leq M_1 \leq M_0 = G$ are two composition series for G, then $r = s$ and there is some permutation π of $\{1, 2, \ldots, r\}$ such that,*

$$\frac{M_{\pi(i)}}{M_{\pi(i)+1}} \cong \frac{N_i}{N_{i+1}}, for 1 \leq i \leq r.$$

▶ **Theorem 4** (Correspondence Theorem see e.g., [21]). *Let $K \trianglelefteq G$ and let $v : G \longrightarrow G/K$ be the canonical map i.e. $v(g) = Kg$ for all g. Then $S \mapsto v(S) = S/K$ is a bijection from the family of all those subgroups S of G which contain K to the family of all the subgroups of G/K.*

Model of computation. In this paper, we use an abstract model of computation known as the word-RAM model. In this model, data is stored in resisters and memory units. Each memory unit and resister can store $O(\log n)$ bits where n is the size of the input. The unit of storage is called *word*. The machine in the word-RAM model can access a word and do the usual arithmetic, logical, and comparison operations in constant time. The input size for our purpose is the order of the group. Without loss of generality, we can assume that the elements of groups are $1, 2, 3, ..., n$. Thus, every group element can be stored in a word and can be accessed in constant time.

There are two phases in the construction of a data structure: *the preprocessing phase* and *the query phase*. In the preprocessing phase, we assume that we have been given a finite group by its Cayley table. Using the Cayley table, we construct a data structure that consists of some arrays and tables. In the query phase, we process multiplication queries. In a multiplication query, two group elements g_1 and g_2 are given by the user. The task is to find the product of g_1 and g_2. In this phase, the data structure constructed in the preprocessing phase is accessed to answer the query. The time taken to answer a single query is called the *query-time*.

The time and space used in preprocessing stage are not considered. We only consider the space used by the data structure and the time it takes to answer a query to multiply the group elements.

▶ **Definition 5.** *Let G be a group and s and t be two positive real numbers. We say that G has an (s,t)-data structure, if G can be stored in a data-structure that uses at most s space and can answer a multiplication query in time at most t.*

▶ **Definition 6.** *Let $\mathcal{G}$ be a class of group and let $s,t : \mathbb{N} \to \mathbb{R}_{\geq 0}$ be two functions. If for every group $G \in \mathcal{G}$ of order n there is a data structure that uses $O(s(n))$ space to store G and can answer a multiplication query in time at most $O(t(n))$ then we say that $\mathcal{G}$ has an $(O(s(n)), O(t(n)))$-data structure.*

3 Extension Theorems

In this section, we discuss how to use data structures for subgroups to build new data structures for groups containing the subgroups.

▶ **Theorem 7.** *There exist positive constants c and d such that for any group G and a subgroup H of G if H has an (s,t)-data structure for some s and t then G has an $(s + c([G:H]^2 + |G|), 2t + d)$-data structure.*

Proof. First we fix a left traversal L and a right traversal R of H in G. Each $g \in G$ can be uniquely written as $g = hr$ where $h \in H$ and $r \in R$. Thus we can define functions $s_R : G \longrightarrow H$ and $c_R : G \longrightarrow R$ such that $g = s_R(g)c_R(g)$. Similarly we can define $c_L : G \longrightarrow L$ and $s_L : G \longrightarrow H$ such that $g = c_L(g)s_L(g)$. We can store these four functions in four arrays each of length $|G|$.

Suppose we need to find the product of g_1 and g_2. Note that,

$$g_1g_2 = c_L(g_1)s_L(g_1)s_R(g_2)c_R(g_2).$$

Since $s_L(g_1), s_R(g_2) \in H$, we can use the data structure for H to find $s_L(g_1)s_R(g_2)$ within time t. Let $h_1 = s_L(g_1)s_R(g_2)$. Therefore, we can write $g_1g_2 = c_L(g_1)h_1c_R(g_2)$.

Given $l \in L$ and $h \in H$, we know that there exist unique elements $h' \in H$ and $r \in R$ such that $lh = h'r$. Thus, we can define two functions $Flip_H : L \times H \longrightarrow H$ and $Flip_R : L \times H \longrightarrow R$ such that $lh = Flip_H(l,h)Flip_R(l,h)$. We can store $Flip_H$ and $Flip_R$ in two 2-dimensional arrays using space linear in $|H \times L| = |G|$. With the help of these functions, we can write

$$g_1g_2 = Flip_H(c_L(g_1), h_1)Flip_R(c_L(g_1), h_1)c_R(g_2) = h_2r_1r_2$$

where $h_2 = Flip_H(c_L(g_1), h_1)$, $r_1 = Flip_R(c_L(g_1), h_1)$ and $r_2 = c_R(g_2)$.

Again we use the fact that any element g of G can be uniquely written as $g = hr$ where $h \in H$ and $r \in R$ to define the functions $Cross_H : R \times R \longrightarrow H$ and $Cross_R : R \times R \longrightarrow R$ such that for all $r, r' \in R$ we have $rr' = Cross_H(r, r')Cross_R(r, r')$. Note that we can store these functions in two 2-dimensional arrays each requiring size linear in $|R \times R| = (|G|/|H|)^2$. With the help of these functions we can write

$$g_1 g_2 = h_2 Cross_H(r_1, r_2) Cross_R(r_1, r_2) = h_2 h_3 r_3$$

where $Cross_H(r_1, r_2) = h_3$ and $r_3 = Cross_R(r_1, r_2)$.

Again we can use the data structure for H to compute the product $h_4 = h_2 h_3$ within time t. Thus $g_1 g_2 = h_4 r_3$. Finally, we define a function $Fuse : H \times R \longrightarrow G$ simply as $Fuse(h, r) = hr$ for all $h \in H$ and $r \in R$. Clearly, a 2-dimensional array to store $Fuse$ would take space linear in $|H \times R| = |G|$. Thus, to produce the final result we just return $g_1 g_2 = Fuse(h_4, r_3)$.

All the functions except for $Cross_R$ and $Cross_H$ take space linear in $|G|$, while $Cross_R$ and $Cross_H$ take space linear in $(|G|/|H|)^2$. The data structure for H takes space at most s. Therefore, the total space required is linear in $|G| + (|G|/|H|)^2$. We note that each function defined in this proof is queried exactly once. Thus, the time to query all the nine functions is bounded by some constant d. Additionally, the time taken to query the data structure for H is at most $2t$. Therefore, we have the required data structure for G. ◀

An immediate corollary of the above theorem is the following.

▶ **Corollary 8.** *Let $0 < c_1 \leq c_2$ be two constants. Let $\mathcal{G}_{c_1,c_2}$ be the class of groups G that has a subgroup H with $c_1\sqrt{|G|} \leq |H| \leq c_2\sqrt{|G|}$. Then $\mathcal{G}_{c_1,c_2}$ has $(O(n), O(1))$ data-structures.*

Proof. The Cayley table for H takes size at most $c_2^2|G|$ and answers queries in constant time. Since $|G|/|H| \leq (1/c_1)\sqrt{|G|}$, we have $(|G|/|H|)^2 \leq (1/c_1)^2|G|$. Hence the result follows from Theorem 7. ◀

In the next theorem we show how to use the data-structure for a normal subgroup of a group to build a data structure for the group when the quotient group is cyclic.

▶ **Theorem 9.** *There are positive constants c and d such that for every group G and any normal subgroup N of G, if G/N is cyclic and N has an (s, t)-data structure for some s and t, then G has an $(s + c|G|, 2t + d)$-data structure.*

Proof. Since G/N is cyclic it is generated by an element g_0N where $g_0 \in G$. The cosets of N in G are $N, g_0N, g_0^2N, \ldots, g_0^{k-1}N$ where k is the order of the group G/N, i.e., $k = [G : N]$. Clearly, $k \leq |G|$. Let $S = \{0, 1, \ldots, k-1\}$.

The set $\{g_0^0, g_0^1, \ldots, g_0^{k-1}\}$ is a left as well as a right traversal of N in G. Hence any element g could be uniquely written as $g = g_0^r n = n' g_0^r$ for some $r \in S$ and $n, n' \in N$. This enables us to define functions $e : G \longrightarrow S$, $s_R : G \longrightarrow N$ and $s_L : G \longrightarrow N$ such that for all $g \in G$

$$g = g_0^{e(g)} s_R(g) = s_L(g) g_0^{e(g)}.$$

These three functions could be stored in arrays each having size $|G|$. To multiply g_1 and g_2 we first observe that $g_1 g_2 = g_0^{e(g_1)} s_R(g_1) s_L(g_2) g_0^{e(g_2)}$. These expression could be obtained by querying each of the functions once. The product $n_1 = s_R(g_1)s_L(g_2)$ can be obtained using the data structure for N within query-time t. Thus $g_1 g_2 = g_0^{\alpha} n_1 g_0^{\beta}$, where $\alpha = e(g_1)$ and $\beta = e(g_2)$.

Next we define a function $Flip : N \times S \longrightarrow N$ with the property that for all $n \in N$ and $i \in S$, $ng_0^i = g_0^i Flip(n, i)$. In other words, $Flip(n, i)$ is just $g_0^{-i} n g_0^i$. This function can be stored in space linear in $|N \times S| = |G|$. Now we can write $g_1 g_2 = g_0^\alpha g_0^\beta Flip(n_1, \beta) = g_0^{\alpha+\beta} n_2$ where $n_2 = Flip(n_1, \beta)$.

Next we compute $g_0^\alpha g_0^\beta = g_0^{\alpha+\beta}$. Observe that $\alpha + \beta \in \{0, 1, \ldots, 2k-2\}$. We define two functions $red_e : \{0, 1, \ldots, 2k-2\} \longrightarrow S$ and $red_N : \{0, 1, \ldots, 2k-2\} \longrightarrow N$ such that $g_0^\ell = g_0^{red_e(\ell)} red_N(\ell)$ for all $\ell \in \{0, \ldots, 2k-2\}$. Note that for $\ell < k$, $red_e(\ell) = \ell$ and $red_N(\ell) = id$. These two functions can be stored using space linear in k. Since $k \leq |G|$, the space required is at most linear in G.

Therefore,

$$g_1 g_2 = g_0^{\alpha+\beta} n_2 = g_0^{red_e(\alpha+\beta)} red_N(\alpha + \beta) n_2.$$

As before the product n_3 of $red_N(\alpha + \beta)$ and n_2 can be found using the data structure for N. Let $red_e(\alpha + \beta) = \gamma$. Hence, $g_1 g_2 = g_0^\gamma n_3$.

We finally define a function $Fuse : S \times N \longrightarrow G$ as $Fuse(i, n) = g_0^i n$ for all $i \in S$ and $n \in N$. Clearly, the function $Fuse$ can be stored using space linear in $|G|$.

The product $g_1 g_2$ is just $Fuse(\gamma, n_3)$.

Each function defined in this proof takes space linear in $|G|$ and the data structure for N takes space at most s. Each function is queried exactly once and the data structure for N is queried twice. This proves the theorem. ◀

4 Compact Data Structures for Finite Groups

Let G be a group of order n. Our goal is to design a constant query-time data structure for G of size linear in n. We first consider a composition series $1 = G_k \lhd \ldots G_1 \lhd G_0 = G$ of G. In case there is a subgroup G_i in the composition series with size within a constant factor of $\sqrt{n}$, we can apply Corollary 8 to obtain a $(O(n), O(1))$ data structure for G. Otherwise we consider the smallest subgroup G_i of order more than $\sqrt{n}$. Note that here $|G_{i+1}|$ is at most $\sqrt{n}$ and therefore G_{i+1} will have its Cayley table of size at most n. This Cayley table can be used to answer a multiplication query involving elements in G_{i+1} in constant time.

Now we consider the composition factor G_i/G_{i+1}. This quotient is a simple group. If this is an abelian group it must be cyclic (of prime order) and we can use Theorem 9 to get a data structure for G_i. Then an application of Theorem 7 with G and its subgroup G_i will give us the required data structure for G.

The nontrivial case is when G_i/G_{i+1} is nonabelian. This is where we use the Classification Theorem of Finite Simple Groups. The classification theorem allows us to split the nonabelian case into various subcases. In each of the subcases we show that we can insert two subgroups G_{i_2} and G_{i_1} such that $G_{i+1} < G_{i_2} < G_{i_1} < G_i$ in such a manner that the indices $[G_{i_2} : G_{i+1}]$, $[G_{i_1} : G_{i_2}]$ and $[G_i : G_{i_1}]$ are all "small". Since G_{i+1} already has a constant query-time data structure (namely its Cayley table) of size linear in n, this allows us to use Theorem 7 successively to the group and subgroup pairs (G_{i_2}, G_{i+1}), (G_{i_1}, G_{i_2}), and (G_i, G_{i_1}) to obtain a constant query-time data structure for G_i of size linear in n. Finally, another application of Theorem 7 with G and its subgroup G_i will give us the required data structure for G.

4.1 Solvable Finite Groups

In this subsection we consider the class $\mathcal{G}_{solv}$ of finite solvable groups. We do this case first before going to the general case for the class of all finite groups because it is independent of the Classification Theorem for Finite Simple Groups.

▶ **Theorem 10.** *The class $\mathcal{G}_{solv}$ has $(O(n), O(1))$ data-structures.*

Proof. Let G be a group and $1 = G_k \lhd \dots G_1 \lhd G_0 = G$ be a composition series of G. Let $n = |G|$.

Case 1: There is i such that $\sqrt{n}/2 \leq |G_i| \leq \sqrt{n}$. We simply apply Corollary 8 to get the desired data structure.

Case 2: There is no i such that $\sqrt{n}/2 \leq |G_i| \leq \sqrt{n}$. Let i be the largest index such that $\sqrt{n} < |G_i|$. We will have $|G_{i+1}| < \sqrt{n}/2$. The Cayley table for G_{i+1} has at most $n/4$ entries. Since G is solvable G_i/G_{i+1} is cyclic of prime order. This allows us to use Theorem 9 to obtain a constant query-time data structure for G_i which is linear in n. Next we observe that $[G : G_i]$ is at most $\sqrt{n}$. If we apply Theorem 7 on G and its subgroup G_i we get the required data structure for G. ◀

4.2 The General Case

Before considering the case for general finite groups we need the following result for nonabelian simple groups.

▶ **Lemma 11.** *There are positive constants b_1 and b_2 such that for any nonabelian simple group H there exist subgroups H_1 and H_2 such that $1 \leq H_2 \leq H_1 \leq H$ and $|H_2| \leq \sqrt{|H|}$, $[H : H_1] \leq b_1\sqrt{|H|}$, and $[H_1 : H_2] \leq b_2\sqrt{|H|}$.*

Proof. The proof uses the Classification Theorem of Finite Simple Group (CFSG). The proof idea is given in Section 5 and the details are given in the Appendix. ◀

Next we prove the main theorem of the paper. We note that Case 2 in the proof of the following theorem can be viewed as a generalized version of the problem of designing linear space and constant query-time data structure for nonableian simple groups.

▶ **Theorem 12.** *The class $\mathcal{G}_{fin}$ of all finite groups has $(O(n), O(1))$ data structures.*

Proof. Let G be a group of order n. We start by considering a composition series $1 = G_k \lhd \dots G_1 \lhd G_0 = G$ be a composition series of G.

Case 1: This is the case when there is i such that $\sqrt{n}/2 \leq |G_i| \leq \sqrt{n}$. This case is exactly similar to the case for solvable groups.

Case 2: As before in this case we assume that there is no composition series element G_i with order more that $\sqrt{n}/2$ but less than $\sqrt{n}$. Let i be the largest index such $\sqrt{n} < |G_i|$. We will then have $|G_{i+1}| < \sqrt{n}/2$. Clearly, the Cayley table of G_{i+1} will have at most $n/4$ entries. Since $[G : G_i] < \sqrt{n}$, by Theorem 7 it is enough to design constant query-time data structure for G_i of size linear in n. In the rest of the proof we therefore concentrate on designing a constant query-time data structure for G_i that uses $O(n)$ space.

If the composition factor G_i/G_{i+1} is abelian then we are again in the same situation as in the second case of solvable groups. Therefore we assume that G_i/G_{i+1} is nonabelian.

We apply Lemma 11 to $H = G_i/G_{i+1}$ to obtain subgroups H_1 and H_2 such that $1 \leq H_2 \leq H_1 \leq H = G_i/G_{i+1}$. By the correspondence theorem of groups, H_1 and H_2 will be of the form G_{i_1}/G_{i+1} and G_{i_2}/G_{i+1} respectively for some subgroups G_{i_1} and G_{i_2} such that $G_{i+1} \leq G_{i_2} \leq G_{i_1} \leq G_i$. From Lemma 11 we have $[H_1 : H_2] \leq b_2\sqrt{|H|}$. Since, $H_1 = G_{i_1}/G_{i+1}$ and $H_2 = G_{i_2}/G_{i+1}$, we have $[G_{i_1}/G_{i+1} : G_{i_2}/G_{i+1}] \leq b_2\sqrt{|G_i/G_{i+1}|} \leq b_2\sqrt{n}$.

Therefore, $[G_{i_1} : G_{i_2}] \leq b_2\sqrt{n}$. Similarly, $[G_i : G_{i_1}] \leq b_1\sqrt{n}$. Again from Lemma 11, we have $H_2 \leq \sqrt{|H|}$. This implies, $[G_{i_2} : G_{i+1}] \leq \sqrt{|G_i/G_{i+1}|} \leq \sqrt{n}$.

Since G_{i+1} has a Cayley table of size at most n and $[G_{i_2} : G_{i+1}] \leq \sqrt{n}$, we will have a constant query-time data structure for the subgroup G_{i_2} of size at most n by Theorem 7. Since $[G_{i_1} : G_{i_2}] \leq b_2\sqrt{n}$ and $[G_i : G_{i_1}] \leq b_1\sqrt{n}$, another two applications of Theorem 7 with the group and subgroup pairs (G_{i_1}, G_{i_2}) and (G_i, G_{i_1}) will give a data structure for G_i of size linear in n which can answer a multiplication query in constant time. ◀

We note that there exist polynomial time algorithms for finding a composition series [22] and checking if a composition factor is abelian [14]. First, we note that G_{i_1} and G_{i_2} can be found simply by a brute force approach. Therefore, we can actually *construct* the data structure for G in the above theorem. While obtaining a polynomial time algorithm to construct the data structure is not our main goal, we note that we can also construct the data structure in polynomial time. The proof of this involves careful use of existing results from group theory and algorithms for group theoretic problems.

5 Proof Sketch for Lemma 11

In this section we sketch the proof idea behind Lemma 11. We first state the Classification Theorem of Finite Simple Groups.

▶ **Theorem 13** ([27]). *(The Classification Theorem of Finite Simple Group) Every finite simple group is isomorphic to one of the following:*

(i) *a cyclic group C_p of prime order p;*

(ii) *an alternating group A_m, for $m \geq 5$;*

(iii) *a classical group;*

- **a.** *linear:* $A_{\mathrm{m}}(q)(or\ \mathrm{PSL}_{\mathrm{m}+1}(q)), \mathrm{m} \geq 1$, *except* $\mathrm{PSL}_2(2)$ *and* $\mathrm{PSL}_2(3)$;
- **b.** *unitary:* ${}^2A_{\mathrm{m}}(q^2)(or \mathrm{PSU}_{\mathrm{m}+1}(q)), \mathrm{m} \geq 2$, *except* $\mathrm{PSU}_3(2)$;
- **c.** *symplectic:* $C_{\mathrm{m}}(q))(or\ \mathrm{PS}_{\mathrm{p}_{2\mathrm{m}}}(q)), \mathrm{m} \geq 2$ *except* $\mathrm{PS}_{p_4}(2)$;
- **d.** *orthogonal:* $B_{\mathrm{m}}(q)(or\ \mathrm{P}\Omega_{2\mathrm{m}+1}(q)), \mathrm{m} \geq 3, q$ *odd;*
  $D_{\mathrm{m}}(q)(or\ \mathrm{P}\Omega^+_{2\mathrm{m}}(q)), \mathrm{m} \geq 4$;
  ${}^2D_{\mathrm{m}}(q^2)(or\ \mathrm{P}\Omega^-_{2\mathrm{m}}(q)), \mathrm{m} \geq 4$

where q is a power p^a of some prime;

(iv) *an exceptional group of Lie type:*

$$\mathrm{G}_2(\mathrm{q}), \mathrm{q} \geq 3; \mathrm{F}_4(\mathrm{q}); \mathrm{E}_6(\mathrm{q}); {}^2\mathrm{E}_6(\mathrm{q}); {}^3\mathrm{D}_4(\mathrm{q}); \mathrm{E}_7(\mathrm{q}); \mathrm{E}_8(\mathrm{q})\ or$$

where q is a power p^a of some prime;

$${}^2\mathrm{B}_2(2^{2\mathrm{m}+1}), \mathrm{m} \geq 1; {}^2\mathrm{G}_2(3^{2\mathrm{m}+1}), \mathrm{m} \geq 1; {}^2\mathrm{F}_4(2^{2\mathrm{m}+1}), \mathrm{m} \geq 1$$

or the Tits group ${}^2F_4(2)'$;

(v) *one of 26 sporadic simple groups:*

- **a.** *the five Mathieu groups* $\mathrm{M}_{11}, \mathrm{M}_{12}, \mathrm{M}_{22}, \mathrm{M}_{23}, \mathrm{M}_{24}$;
- **b.** *the seven Leech Lattice groups* $\mathrm{Co}_1, \mathrm{Co}_2, \mathrm{Co}_3, \mathrm{McL}, \mathrm{HS}, \mathrm{Suz}, \mathrm{J}_2$;
- **c.** *the three Fischer groups* $\mathrm{Fi}_{22}, \mathrm{Fi}_{23}, \mathrm{Fi}'_{24}$;
- **d.** *the Monstrous groups* $\mathbb{M}, \mathbb{B}, \mathrm{Th}, \mathrm{HN}, \mathrm{He}$;
- **e.** *the six pariahs* $\mathrm{J}_1, \mathrm{J}_2, \mathrm{J}_4, \mathrm{O'N}, \mathrm{Ly}, \mathrm{Ru}$.

The definition of each of the group classes mentioned in the above theorem can be found in the standard texts on CFSG (see e.g., [6], [27], [3]).

Since Lemma 11 is about nonabelian simple groups we need to consider cases (ii) to (v) in Theorem 13. We take each subcases under these cases and show that there are subgroups H_1 and H_2 satisfying the conditions of the lemma.

We note that the 26 sporadic simple groups listed in the case (v) are of constant sizes. Therefore, we can ignore these groups for the purpose of the proof by simply taking H_2 to be the identity subgroup and H_1 to be H. Of course if we do so we need to pick extremely large constant b_2 as some the sporadic simple groups are of huge sizes. Fortunately, there are known results on the groups listed under case (v) that helps us to keep the constants b_1 and b_2 under 5.

We handle the *Alternating group* (case (ii) of Theorem 13) case as follows. Notice that one can find $k \in \mathbb{Z}$ such that $\frac{k!}{2} \leq \sqrt{\frac{m!}{2}} < \frac{(k+1)!}{2}$, and $H_2 \cong A_k$ and $H_1 \cong A_{k+1}$. Then clearly, $|H_2|^2 = \left(\frac{k!}{2}\right)^2 \leq \frac{m!}{2}$. The inequality $\frac{m!}{2} < \left(\frac{(k+1)!}{2}\right)^2$ implies that $k > \frac{m}{2}$. The value of k can be computed easily. One can also check that, $\left(\frac{|H_1|}{|H_2|}\right)^2 = (k+1)^2 \leq \frac{m!}{2}$ and $\left(\frac{|H|}{|H_1|}\right)^2 < \frac{m!}{2}$.

For the remaining groups we use the following two methods for the choices of H_1 and H_2. The methods are as follows:

1. **Method 1:** In this method, we first choose H_2 to be a certain Sylow subgroup of the given simple group H. Next we pick H_1 to be the normalizer of H_2 in H or the Borel subgroup containing H_2..

   Example: Let us take H to be a simple group $A_m(q)$ for some $q > 2$ which appears in case (iii) of Theorem 13. Here q is power of some prime p. It is known that $A_m(q)$ has order $q^{m(m+1)/2} \prod_{i=1}^{m} (q^{i+1} - 1)/(q - 1, m + 1)$ where $(q - 1, m + 1)$ denotes the gcd of $q - 1$ and $m + 1$ (see [3], p. 252). Clearly, H will have a Sylow p-subgroup of order $q^{m(m+1)/2}$. We set H_2 to be this subgroup. Next we pick H_1 to be the normailzer of H_2 in H. It is also known that the order of H_1 is $q^{m(m+1)/2}(q - 1)^m$ (see [27], p. 46). One can check that with $b_1 = 2$ and $b_2 = 1$, these choices satisfy the conditions of Lemma 11 (see Appendix for the details).

2. **Method 2:** In this method, we choose H_1 to be a maximal subgroup of the simple group H and H_2 to certain Sylow subgroup of H_1.

   Example: In the example under Method 1 we consider the case for $A_m(q)$ when $q > 2$. In this example we take the case when $q = 2$. Here $H = A_m(q)$ will have order $2^{m(m+1)/2} \prod_{i=1}^{m} (2^{i+1} - 1)$ (see [3], p. 252). It is known that the maximal subgroup of H is of order $|H|/(2^m - 1)$ (see [16], p. 175). We take this subgroup as H_1. Next we take H_2 as a Sylow 2-subgroup of H_1 which has order $2^{m(m+1)/2}$. It is easy to verify that these choices of H_1 and H_2 along with $b_1 = b_2 = 1$ satisfy the conditions of Lemma 11 (see Appendix for the details).

Table 1 lists the methods that we have used for choosing the suitable subgroups in the corresponding nonabelian simple group. The last two columns represent the constant factors b_1 and b_2 for the corresponding simple group (see Table 1).

For case (v), we use Method 2 to get the suitable subgroups.

Appendix A.3 contains two comprehensive tables listing the orders of subgroups used in the proof of Lemma 11 for different cases of CFSG.

Table 1 Table representing the constant factor and method used for choosing suitable subgroups.

Case	H	Condition on q	Method	b_1	b_2
(iii)	$A_m(q)$	$q > 2$	Method 1	2	1
		$q = 2$	Method 2	1	1
	${}^2A_m(q^2); m > 1$	$q > 2$	Method 1	2	1
		$q = 2; 6 \nmid (m-1)$	Method 2	1	1
		$q = 2; 6 \mid (m-1)$	Method 2	1	1
	$C_m(q); m > 2$	$q > 2$	Method 1	2	1
		$q = 2$	Method 2	1	1
	$B_m(q); m > 1$	q odd	Method 1	2	1
	$D_m(q); m > 3$	$q > 2$	Method 1	2	1
		$q = 2$	Method 2	1	1
	${}^2D_m(q^2); m > 3$	$q > 2$	Method 1	3	1
		$q = 2$	Method 2	1	1
(iv)	$G_2(q)$	$q \geq 3$	Method 1	1	1
	$F_4(q)$	All q	Method 2	1	1
	$E_6(q)$	$q > 2$	Method 1	1	1
		$q = 2$	Method 2	1	1
	${}^2E_6(q)$	All q	Method 1	1	1
	${}^3D_4(q)$	All q	Method 1	1	1
	$E_7(q)$	$q > 2$	Method 1	1	1
		$q = 2$	Method 2	1	1
	$E_8(q)$	$q > 2$	Method 1	1	1
		$q = 2$	Method 2	1	1
	${}^2B_2(q)$	$q = 2^{2t+1}, t \geq 1$	Method 1	1	1
	${}^2G_2(q)$	$q = 3^{2t+1}, t \geq 1$	Method 1	1	1
	${}^2F_4(q)$	$q = 2^{2t+1}, t \geq 1$	Method 1	1	1
	${}^2F_4(2)'$	$q = 2$	Method 2	1	1

References

1 Vikraman Arvind and Jacobo Torán. The complexity of quasigroup isomorphism and the minimum generating set problem. In *International Symposium on Algorithms and Computation*, pages 233–242. Springer, 2006.

2 Vikraman Arvind and Jacobo Torán. Solvable group isomorphism is (almost) in NP ∩ conp. *ACM Trans. Comput. Theory*, 2(2):4:1–4:22, 2011. doi:10.1145/1944857.1944859.

3 M. Aschbacher. *Finite Group Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2000. doi:10.1017/CBO9781139175319.

4 László Babai, Paolo Codenotti, and Youming Qiao. Polynomial-time isomorphism test for groups with no abelian normal subgroups. In *International Colloquium on Automata, Languages, and Programming*, pages 51–62. Springer, 2012.

5 László Babai and Youming Qiao. Polynomial-time isomorphism test for groups with abelian sylow towers. In *STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)*, volume 14, pages 453–464. LIPIcs, 2012.

6 Roger W. Carter. *Finite groups of Lie type*. Wiley Classics Library. John Wiley & Sons, Ltd., Chichester, 1993. Conjugacy classes and complex characters, Reprint of the 1985 original, A Wiley-Interscience Publication.

7 J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker, and R. A. Wilson. *ATLAS of Finite Groups*. Oxford University Press, Eynsham, 1985. Maximal subgroups and ordinary characters for simple groups, With computational assistance from J. G. Thackray.

8 Bireswar Das and Shivdutt Sharma. Compact data structures for dedekind groups and finite rings. In *WALCOM*, pages 90–102, 2021.

9 Bireswar Das, Shivdutt Sharma, and P. R. Vaidyanathan. Space efficient representations of finite groups. *J. Comput. Syst. Sci.*, 114:137–146, 2020. doi:10.1016/j.jcss.2020.06.007.

10 David S. Dummit and Richard M. Foote. *Abstract algebra*. John Wiley & Sons, Inc., Hoboken, NJ, third edition, 2004.

11 Arash Farzan and J. Ian Munro. Succinct representation of finite abelian groups. In *ISSAC 2006*, pages 87–92. ACM, New York, 2006.

12 Merrick Furst, John Hopcroft, and Eugene Luks. Polynomial-time algorithms for permutation groups. In *21st Annual Symposium on Foundations of Computer Science (sfcs 1980)*, pages 36–41. IEEE, 1980.

13 François Le Gall. Efficient isomorphism testing for a class of group extensions. *CoRR*, abs/0812.2298, 2008. arXiv:0812.2298.

14 T. Kavitha. Linear time algorithms for abelian group isomorphism and related problems. *J. Comput. System Sci.*, 73(6):986–996, 2007.

15 Neeraj Kayal and Timur Nezhmetdinov. Factoring groups efficiently. In *International colloquium on automata, languages, and programming*, pages 585–596. Springer, 2009.

16 Peter B. Kleidman and Martin W. Liebeck. *The Subgroup Structure of the Finite Classical Groups*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1990. doi:10.1017/CBO9780511629235.

17 Daniel J. Kleitman, Bruce R. Rothschild, and Joel H. Spencer. The number of semigroups of order n. *Proc. Amer. Math. Soc.*, 55(1):227–232, 1976.

18 S Ravi Kumar and Ronitt Rubinfeld. Property testing of abelian group operations, 1998.

19 Gary L. Miller. On the $n^{\log n}$ isomorphism technique: A preliminary report. In Richard J. Lipton, Walter A. Burkhard, Walter J. Savitch, Emily P. Friedman, and Alfred V. Aho, editors, *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1-3, 1978, San Diego, California, USA*, pages 51–58. ACM, 1978.

20 Youming Qiao, Jayalal Sarma, and Bangsheng Tang. On isomorphism testing of groups with normal hall subgroups. *J. Comput. Sci. Technol.*, 27(4):687–701, 2012. doi:10.1007/s11390-012-1255-7.

21 Joseph J. Rotman. *An introduction to the theory of groups*, volume 148 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, fourth edition, 1995.

22 Ákos Seress. *Permutation group algorithms*, volume 152 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2003.

23 Charles C. Sims. Computational methods in the study of permutation groups. In John Leech, editor, *Computational Problems in Abstract Algebra*, pages 169–183. Pergamon, 1970. doi:10.1016/B978-0-08-012975-4.50020-5.

24 Charles C Sims. Computation with permutation groups. In *Proceedings of the second ACM symposium on Symbolic and algebraic manipulation*, pages 23–28, 1971.

25 Charles C. Sims. *Computation with finitely presented groups*, volume 48 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1994.

26 J. H. van Lint and R. M. Wilson. *A course in combinatorics*. Cambridge University Press, Cambridge, 1992.

27 Robert A. Wilson. *The finite simple groups*, volume 251 of *Graduate Texts in Mathematics*. Springer-Verlag London, Ltd., London, 2009. doi:10.1007/978-1-84800-988-2.

28 Robert A. Wilson. Maximal subgroups of sporadic groups. In *Finite simple groups: thirty years of the atlas and beyond*, volume 694 of *Contemp. Math.*, pages 57–72. Amer. Math. Soc., Providence, RI, 2017.

A Proof of Lemma 11

In this section we indicate how to prove Lemma 11 in more detail. We do this in the ordering mentioned in the Classification Theorem of Finite Simple Group, i.e., Theorem 13. As we mentioned in Section 2, we just need to use some known results on the *order* of certain subgroups of simple groups. The detailed description of these groups may be skipped for the purpose of the proof. The results that are used in the proof are on the orders of the finite simple groups, on the orders of maximal subgroups of simple groups and the normalizers of certain types of Sylow subgroups of simple groups. The information about the order of these simple groups can be obtained from [3]

In case (ii) of Theorem 13, H is an alternating group. We have already seen that the subgroups H_2 and H_1 are certain suitably picked stabilizer subgroups of the given alternating group H.

For the cases (iii) and (iv) of Theorem 13, we use Method 1 and Method 2 to get the desired subgroups as required in Lemma 11. In this cases the finite simple group H is of Lie-type and is defined over a finite field $\mathbb{F}_q$ where q is a power of some prime p. In Method 1, we take H_2 to be certain Sylow p-subgroup of H. The existence of such H_2 follows from the well-known Sylow theorem. For the existence of H_1, we take the normalizer of H_2 or the Borel subgroup. The information about the order of normalizer has been obtained from (see [6], p. 76, [27], p. 46).

For the groups in which we use Method 2, we consider a maximal subgroup of H as H_1 and H_2 to be some Sylow p-subgroup of H_1. The index of a maximal subgroup (and hence its order) can be obtained from [16], p. 175 and [27], p. 156.

For the simple groups in case (v), we use Method 2 and the information about order of maximal subgroup (H_1) can be obtained from [28]. Also, for the choice of H_2, we choose certain Sylow subgroup of H_1.

The inequalities in the following two remarks are used in the calculation multiple times.

▶ Remark 14. For all integer $q > 2$, we have $\frac{q}{(q-1)^2} < 1$.

▶ Remark 15. $\prod_{i=1}^{i=m}(q^{i+1} - (-1)^{i+1}) < q^{\sum_{i=1}^{i=m}(i+1)}$.

▶ Remark 16. The gcd of two natural numbers m and n is denoted by (m, n).

A.1 The Classical Groups

We have seen the case(ii) of Theorem 13 in Section 5. In this section, we consider H to be a classical simple group described in case (iii) of Theorem 13. In particular, we consider the case when H is ${}^2A_m(q^2)$ where q is a power of some prime p. All the other cases can be handle similarly. As described earlier, we use Method 1 and Method 2 to show the existence of subgroups H_2 and H_1 of the simple group H.

1.1 $H = {}^2A_m(q^2)$; $m \geq 2$, $q > 2$ (Method 1)

The finite simple group ${}^2A_m(q^2)$ is isomorphic to the *projective special unitary group* $\mathrm{PSU}_{m+1}(q)$. The group $\mathrm{PSU}_{m+1}(q)$ is the group obtain by taking special unitary group $\mathrm{SU}_{m+1}(q)$ and quotienting it by its center, i.e. ${}^2A_m(q^2) \cong \frac{\mathrm{SU}_{m+1}(q)}{Z(\mathrm{SU}_{m+1}(q))}$ (see [27], p. 66). It is known that (see [3], p. 252) the order of ${}^2A_m(q^2)$ is,

$$|H| = \frac{q^{\frac{m(m+1)}{2}}}{(q+1, m+1)} \prod_{i=1}^{m}(q^{i+1} - (-1)^{i+1}).$$

Let H_2 be the Sylow p-subgroup of ${}^2A_m(q^2)$, then $|H_2| = q^{\frac{m(m+1)}{2}}$ and $|H_2|^2 \leq |H|$. Let H_1 be the Borel subgroup of H of order (see [27], [6]),

$$|H_1| = \frac{q^{\frac{m(m+1)}{2}}}{(q+1,m+1)}(q-1)^{\lfloor m/2 \rfloor}(q+1)^{\lceil \frac{m-1}{2} \rceil}.$$

One can check that $\left(\frac{|H_1|}{|H_2|}\right)^2 \leq |H_1| < |H|$ and $\frac{|H|}{|H_1|} \leq 2\sqrt{|H|}$.

1.2 $H = {}^2A_m(q^2)$; $m \geq 2$, $q = 2$ (Method 2)

The finite simple group ${}^2A_m(2^2)$ is of order $2^{\frac{m(m+1)}{2}} \prod_{i=1}^{m}(2^{i+1} - (-1)^{i+1})/(3, m+1)$ and is isomorphic to *projective special unitary group* $\mathrm{PSU}_{m+1}(2)$ or $\mathrm{U}_{m+1}(q)$. The group $\mathrm{U}_{m+1}(q)$ has a maximal subgroup of index $\frac{(2^{m+1}-(-1)^{m+1})(2^m-(-1)^m)}{3}$ when $6 \nmid (m-1)$ and of index $\frac{2^m(2^{m+1}-1)}{3}$, when $6 \mid (m-1)$ (see [16], p. 175) .

(Case 1) $6 \nmid (m-1)$

Let H_1 be corresponding maximal subgroup of ${}^2A_m(2^2)$ whose index is

$$\frac{(2^{m+1} - (-1)^{m+1})(2^m - (-1)^m)}{3}$$

in ${}^2A_m(2^2)$. Then, the order of H_1 is,

$$|H_1| = \frac{3}{(3, m+1)} \frac{2^{\frac{m(m+1)}{2}} \prod_{i=1}^{m}(2^{i+1} - (-1)^{i+1})}{(2^{m+1} - (-1)^{m+1})(2^m - (-1)^m)}.$$

Let H_2 be the Sylow 2-subgroup of H_1. Then, $|H_2| = 2^{\frac{m(m+1)}{2}}$ and $|H_2|^2 < |\,{}^2A_m(2^2)|$.

It is easy to see that $\frac{\left(\frac{|H_1|}{|H_2|}\right)^2}{|\,{}^2A_m(2^2)|} < 1$ and $\left(\frac{|\,{}^2A_m(2^2)|}{|H_1|}\right)^2 < |\,{}^2A_m(2^2)|$.

(Case 2) $6|(m-1)$ (i.e. $m \geq 7$)

In this case, as we know that the group ${}^2A_m(q^2)$ has a maximal subgroup of index $\frac{2^m(2^{m+1}-1)}{3}$. Let H_1 be one such maximal subgroup. Then,

$$|H_1| = \frac{3}{(3, m+1)} 2^{\frac{m(m-1)}{2}} \prod_{i=1}^{m-1}(2^{i+1} - (-1)^{i+1}).$$

Let H_2 be the Sylow 2-subgroup of H_1, then H_2 has order $2^{\frac{m(m-1)}{2}}$ and $|H_2|^2 < |\,{}^2A_m(2^2)|$.

Clearly we can check that $\left(\frac{|H_1|}{|H_2|}\right)^2 < |\,{}^2A_m(2^2)|$ and $\left(\frac{|\,{}^2A_m(2^2)|}{|H_1|}\right)^2 < |\,{}^2A_m(2^2)|$.

A.2 Exceptional Group of Lie Type

In this section, we consider H to be an exceptional simple group of Lie Type described in case (iv) of Theorem 13. In particular, we consider the case when H is $F_4(q)$, $E_6(q)$ and ${}^2F_4(2)'$ where q is a power of some prime p. The similar arguments can be used to prove the remaining cases.

(1) $H = F_4(q)$ (Method 2)

The finite simple group $F_4(q)$ has order (see [3], p. 252),

$$|F_4(q)| = q^{24}(q^{12}-1)(q^8-1)(q^6-1)(q^2-1).$$

It is known that (see [27], p. 156) the group $F_4(q)$ has a maximal subgroup $q^{1+14} : Sp_6(q).C_{q-1}$ of order $q^{24}(q^6-1)(q^4-1)(q^2-1)(q-1)$ say H_1. This subgroup has a Sylow p-subgroup say H_2 of order q^{24} and $|H_2|^2 \leq |F_4(q)|$. Therefore, $\left(\frac{|H_1|}{|H_2|}\right)^2 < |H_1| < |H|$ and $\left(\frac{|H|}{|H_1|}\right)^2 < |H|$.

(2) $H = E_6(q)$; $q > 2$ (Method 1)
The group $E_6(q)$ is a finite simple group. The order of $H = E_6(q)$ is (see [3], p. 252),

$$|E_6(q)| = \frac{q^{36}}{(3, q-1)}(q^{12}-1)(q^9-1)(q^8-1)(q^6-1)(q^5-1)(q^2-1).$$

Clearly, it has a Sylow p-subgroup H_2 of order q^{36} and $|H_2|^2 \leq |E_6(q)|$. Let H_1 be the Borel subgroup of H then the order of H_1 is $q^{36}(q-1)^6$ (see [27], [6]). Clearly, $\left(\frac{|H_1|}{|H_2|}\right)^2 < |H_1| < |H|$ and $\left(\frac{|H|}{|H_1|}\right)^2 \leq |H|$.
Notice that, the group $E_6(2)$ is of constant order. However, we can use Method 2 to reduce the constants b_1, b_2 to 1. By taking H_1 to be maximal subgroup of order (see [7]) $2^{36} \cdot 3^3 \cdot 5 \cdot 7 \cdot 31$ and H_2 to be its Sylow 2-subgroup of order 2^{36}.

(3) $H = {}^2F_4(2)'$; (Method 2)
The simple group $H = {}^2F_4(2)'$; has order 17971200. It is known that H has a maximal subgroup of order 11232. We take H_1 to be this maximal subgroup and H_2 to be the Sylow 2-subgroup of H_1 which has order 32. Thus, we get $b_1 = b_2 = 1$.

A.3 Tables

In this section we cover the details of Sporadic simple groups (Table 2), and the order of all the simple groups that we define in cases (ii)-(iv) of Theorem 13 in Table 3 and 4. These tables also contain the order of the subgroups H_2 and H_1.
Table 2 represents the information about the subgroups H_2 and H_1 of the Sporadic simple groups. In Table 2 we consider the values of $t_i, i = 1, 2, 3, 4$ as follows.

$t_1 = 808017424794512875886459904961710757005754368000000000$
$t_2 = 2^{42} \cdot 3^{13} \cdot 5^6 \cdot 7^2 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 31 \cdot 47$
$t_3 = 4154781481226426191177580544000000$
$t_4 = 2^{38} \cdot (2^{12}-1) \cdot (2^9+1) \cdot (2^8-1) \cdot (2^6-1) \cdot (2^5+1) \cdot (2^2-1)$

In the Table 3, the values of $c_{mi}, i = 1, 2, 3, 4, 5$ are as follows.

$c_{m1} = \frac{q^{\frac{m(m+1)}{2}}}{(q+1,m+1)}\ (q-1)^{\lfloor m/2 \rfloor}(q+1)^{\lceil \frac{m-1}{2} \rceil}$.
$c_{m2} = \frac{3}{(3,m+1)} \frac{2^{\frac{m(m+1)}{2}} \prod_{i=1}^{m}(2^{i+1}-(-1)^{i+1})}{(2^{m+1}-(-1)^{m+1})(2^m-(-1)^m)}$
$c_{m3} = \frac{3}{(3,m+1)} 2^{\frac{m(m-1)}{2}} \prod_{i=1}^{m-1}(2^{i+1}-(-1)^{i+1})$
$c_{m4} = 2^{m^2-m+1}(2^m+1)\prod_{i=1}^{m-1}(2^{2i}-1)$
$c_{m5} = 2^{m(m-1)}(2^{m-1}+1)\prod_{i=1}^{m-2}(2^{2i}-1)$.

Table 2 Table representing the constant factor and Method used for choosing suitable subgroups.

H	Order of H	Order of H_2	Order of H_1	b_1	b_2
M_{11}	7920	2^4	720	1	1
M_{12}	95040	2^2	660	1	1
M_{22}	443520	2^6	20160	1	1
M_{23}	10200960	2^7	443520	1	1
M_{24}	244823040	2^8	887040	1	1
Co_1	4157776806543360000	262144	42305400000000	1	1
Co_2	42305400000000	262144	908328960	1	1
Co_3	495767000000	2^7	10200960	1	1
McL	898128000	3^6	$3^6 \cdot 2^7 \cdot 7 \cdot 5$	1	1
HS	44352000	2^7	$2^7 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11$	1	1
Suz	448345497600	2^{12}	251596800	1	1
J_2	604800	2^5	6048	1	1
Fi_{22}	64561751654400	2^{16}	$2^{16}(2^6-1)(2^5+1)(2^4-1)(2^3+1)$	1	1
Fi_{23}	4089470473293004800	2^{18}	$2^{18} \cdot 3^9 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13$	1	1
Fi'_{24}	1255205709190661721292800	2^{19}	$2^{19} \cdot 3^{13} \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 23$	1	1
$\mathbb{M}$	t_1	2^{42}	t_2	1	1
$\mathbb{B}$	t_3	2^{38}	t_4	1	1
Th	90745943887872000	2^{15}	319979520	1	1
HN	273030912000000	2^9	239500800	1	1
He	4030387200	2^8	$2^8 \cdot 255 \cdot 15$	1	1
J_1	175560	2^2	660	1	1
J_3	50232960	2^5	8160	1	1
J_4	86775571046077562880	2097152	57161637225	1	1
O'N	460815505920	2^6	3753792	1	1
Ly	51765179004000000	15625	5859000000	1	5
Ru	145926144000	2^{12}	35942400	1	1

Table 3 Order of the simple groups (case (iii) of Theorem 13) and order of its subgroups H_2, H_1.

H	$\lvert H\rvert$	$\lvert H_2\rvert$	$\lvert H_1\rvert$
$A_m(q)$; $q > 2$	$\frac{q^{\frac{m(m+1)}{2}}\prod_{i=1}^{m}(q^{i+1}-1)}{(q-1,m+1)}$	$q^{\frac{m(m+1)}{2}}$	$\frac{q^{\frac{m(m+1)}{2}}}{(q-1,m+1)}(q-1)^m$
$A_m(2)$; $q = 2$	$2^{\frac{m(m+1)}{2}}\prod_{i=1}^{m}(2^{i+1}-1)$	$2^{\frac{m(m+1)}{2}}$	$\frac{2^{\frac{m(m+1)}{2}}\prod_{i=1}^{m}(2^{i+1}-1)}{(2^{m+1}-1)}$
${}^2A_m(q^2)$; $q > 2$, $m > 1$	$\frac{q^{\frac{m(m+1)}{2}}}{(q+1,m+1)}\prod_{i=1}^{m}(q^{i+1}-(-1)^{i+1})$	$q^{\frac{m(m+1)}{2}}$	c_{m1}
${}^2A_m(2^2)$; $q = 2$, $m > 1$, $6 \nmid (m-1)$	$\frac{2^{\frac{m(m+1)}{2}}}{(3,m+1)}\prod_{i=1}^{m}(2^{i+1}-(-1)^{i+1})$	$2^{\frac{m(m+1)}{2}}$	c_{m2}
${}^2A_m(2^2)$; $q = 2$, $m > 1$, $6 \mid (m-1)$	$\frac{2^{\frac{m(m+1)}{2}}}{(3,m+1)}\prod_{i=1}^{m}(2^{i+1}-(-1)^{i+1})$	$2^{\frac{m(m-1)}{2}}$	c_{m3}
$C_m(q)$; $q > 2$, $m > 2$	$\frac{q^{m^2}\prod_{i=1}^{m}(q^{2i}-1)}{(2,q-1)}$	q^{m^2}	$\frac{q^{m^2}}{(2,q-1)}(q-1)^m$
$C_m(2)$; $q = 2$, $m > 2$	$2^{m^2}\prod_{i=1}^{m}(2^{2i}-1)$	2^{m^2-m+1}	c_{m4}
$B_m(q)$; q odd, $m > 1$	$\frac{q^{m^2}\prod_{i=1}^{m}(q^{2i}-1)}{(2,q-1)}$	q^{m^2}	$\frac{q^{m^2}}{(2,q-1)}(q-1)^m$
$D_m(q)$; $q > 2$, $m > 3$	$\frac{q^{m(m-1)}(q^m-1)\prod_{i=1}^{m-1}(q^{2i}-1)}{(4,q^m-1)}$	$q^{m(m-1)}$	$\frac{q^{m(m-1)}}{(4,q^m-1)}(q-1)^m$
$D_m(2)$; $q = 2$, $m > 3$	$2^{m(m-1)}(2^m-1)\prod_{i=1}^{m-1}(2^{2i}-1)$	2^{m^2-2m+1}	$2^{m^2-2m+1}\prod_{i=1}^{m-1}(2^{2i}-1)$
${}^2D_m(q^2)$; $q > 2$, $m > 3$	$\frac{q^{m(m-1)}(q^m+1)}{(4,q^m+1)}\prod_{i=1}^{m-1}(q^{2i}-1)$	$q^{m(m-1)}$	$\frac{q^{m(m-1)}}{(4,q^n+1)}(q-1)^m$
${}^2D_m(2^2)$; $q = 2$, $m > 3$	$\frac{2^{m(m-1)}(2^m+1)}{(4,2^m+1)}\prod_{i=1}^{m-1}(2^{2i}-1)$	$2^{m(m-1)}$	c_{m5}

Table 4 Order of the simple groups (case (iv) of Theorem 13) and order of its subgroups H_2, H_1.

$G_2(q)$	$q^6(q^6-1)(q^2-1)$	q^6	$q^6(q-1)^2$
$F_4(q)$	$q^{24}\prod_{i\in\{2,6,8,12\}}(q^i-1)$	q^{24}	$q^{24}\prod_{i\in\{1,2,4,6\}}(q^i-1)$
$E_6(q)$, $q > 2$	$\frac{q^{36}}{(3,q-1)}\prod_{i\in\{2,5,6,8,9,12\}}(q^i-1)$	q^{36}	$q^{36}(q-1)^6$
$E_6(2)$	$2^{36}\prod_{i\in\{2,5,6,8,9,12\}}(2^i-1)$	2^{36}	$2^{36}\cdot 3^3\cdot 5\cdot 7\cdot 31$
${}^2E_6(q)$	$\frac{q^{36}(q^9+1)}{(3,q+1)}\prod_{i\in\{2,5,6,8,12\}}(q^i-1)$	q^{36}	$q^{36}(q-1)^4(q+1)^2$
${}^3D_4(q)$	$q^{12}(q^8+q^4+1)(q^6-1)(q^2-1)$	q^{12}	$q^{12}\ (q^3-1)(q-1)$
$E_7(q)$, $q > 2$	$\frac{q^{63}}{(2,q-1)}\prod_{i\in\{2,6,8,10,12,14,18\}}(q^i-1)$	q^{63}	$q^{63}(q-1)^7$
$E_7(2)$	$2^{63}\prod_{i\in\{2,6,8,10,12,14,18\}}(2^i-1)$	2^{63}	$2^{63}\cdot 3^4\cdot 7^2\cdot 5$
$E_8(q)$, $q > 2$	$q^{120}\prod_{i\in\{2,8,12,14,18,20,24,30\}}(q^i-1)$	q^{120}	$q^{120}(q-1)^8$
$E_8(2)$	$2^{120}\prod_{i\in\{2,8,12,14,18,20,24,30\}}(2^i-1)$	2^{119}	$2^{119}\cdot 3^4\cdot 5\cdot 7^2\cdot 31$
${}^2B_2(q)$; $q = 2^{2t+1}$, $t \geq 1$	$q^2(q^2+1)(q-1)$	q^2	$q^2(q-1)$
${}^2G_2(q)$; $q = 3^{2t+1}$, $t \geq 1$	$q^3(q^3+1)(q-1)$	q^3	$q^3(q-1)$
${}^2F_4(q)$; $q = 2^{2t+1}$, $t \geq 1$	$q^{12}(q^6+1)(q^4-1)(q^3+1)(q-1)$	q^{12}	$q^{12}(q-1)^2$