SIAM J. COMPUT
Vol. 26, No. 5, pp. 1310–1342, October 1997

003

FAST MANAGEMENT OF PERMUTATION GROUPS I*

LÁSZLÓ BABAI†, EUGENE M. LUKS‡, AND ÁKOS SERESS§

Abstract. We present new algorithms for permutation group manipulation. Our methods result in an improvement of nearly an order of magnitude in the worst-case analysis for the fundamental problems of finding strong generating sets and testing membership. The normal structure of the group is brought into play even for such elementary issues. An essential element is the recognition of large alternating composition factors of the given group and subsequent extension of the permutation domain to display the natural action of these alternating groups. Further new features include a novel fast handling of alternating groups and the sifting of defining relations in order to link these and other analyzed factors with the rest of the group. The analysis of the algorithm depends on the classification of finite simple groups. In a sequel to this paper, using an enhancement of the present method, we shall achieve a further order of magnitude improvement.

Key words. permutation group algorithm, strong generating set

AMS subject classifications. 68Q40, 20B40

PII. S0097539794229417

1. Introduction.

Since the size of a permutation group G on n letters can be exponential in n, it is customary, for computational purposes, to specify G by a small list of generators. However, the succinctness of such a representation raises the issue of whether we can deal effectively with the groups that we can specify. Can one, for example, find the order of G and test membership in G without enumerating all of its elements?

In fact, in the late sixties, Sims developed efficient algorithms for permutation group manipulation [Si70]. These included the key notion of a *strong generating set* (SGS) which is the underlying concept in essentially all polynomial-time algorithms in computational group theory. Given a chain $G = G_0 \geq G_1 \geq \cdots \geq G_m = 1$ of subgroups of G, an SGS with respect to this chain is a set $T \subset G$ such that $T \cap G_i$ generates G_i for each i. Sims's algorithm uses the point stabilizer chain; that is, G_i is the pointwise stabilizer of the first i points of the permutation domain.

While Sims's methods for constructing an SGS have been widely used in computational group theory since their inception, the question of their asymptotic efficiency was not resolved until 1980. Furst, Hopcroft, and Luks [FHL] observed that a version of Sims's algorithm runs in polynomial time, namely $O(n^6 + sn^2)$ steps, where s is the number of generators given for G. Subsequently, Knuth [Kn] and Jerrum [Je82], [Je86] gave variants with running time $O(n^5 + sn^2)$. All of these algorithms rest on the most elementary group theory.

*Received by the editors October 28, 1994; accepted for publication (in revised form) September 20, 1995.

http://www.siam.org/journals/sicomp/26-5/22941.html

†Department of Algebra, Eötvös University, Muzeum krt. 6-8, Budapest H-1088, Hungary and Department of Computer Science, University of Chicago, 1100 East 58th Street, Chicago, IL 60637-1504 (laci@cs.uchicago.edu). The research of this author was partially supported by NSF grant CCR-9014562 and OTKA (Hungary) grant 2581.

‡Computer and Information Sciences Department, University of Oregon, Eugene, OR 97403 (luks@cs.uoregon.edu). The research of this author was partially supported by NSF grant CCR-9013410.

§Department of Mathematics, Ohio State University, 231 West 18th Street, Columbus, OH 43210 (akos@math.ohio-state.edu). The research of this author was partially supported by NSF grants CCR-9201303 and CCR-9503430.

Since Knuth's note of 1981 (a preliminary version of [Kn]) and Jerrum's 1982 paper, the $O(n^5)$ bound has achieved notoriety and is generally believed to be the best that can be obtained via Sims's approach alone (cf. Remark 2.13). The main result of this paper is the improvement of the worst case bound by nearly one order of magnitude.

THEOREM 1.1. *Given a permutation group G by a list S of generators, $|S| = s$, the following problems can be solved in $O(n^4 \log^{c_1} n + sn^2)$ time.*

(a) *Find a set of strong generators.*

(b) *Find the order of G.*

(c) *Test membership of any permutation in G. Additional tests cost only $O(n^2)$ each.*

(d) *Find the pointwise stabilizer of a subset of the permutation domain.*

(e) *Construct a generator-relation presentation $\langle X|R\rangle$ of G in which $|X| = O(n \log^{c_2} n)$ and $|R| = O(n^2 \log^{c_3} n)$.*

In order to avoid long timing expressions as in Theorem 1.1 and concentrate on the essential part of the improvements, we introduce a "soft version" of the big-O notation. For two functions $f(n), g(n)$, we write $f(n) = O^{\sim}(g(n))$ if $f(n) \leq Cg(n) \log^c n$ (c, C are positive constants). Thus the time bound for basic permutation group manipulation in Theorem 1.1 is $O^{\sim}(n^4 + sn^2)$. We do not try, at this time, to minimize the exponent of $\log n$. Straightforward estimates give $c_1 = 7, c_2 = 2$, and $c_3 = 4$.

The new algorithms are not merely improved versions of previous SGS constructions. All of those predecessors construct an SGS with respect to the chain of point stabilizer subgroups. A key departure from the traditional approach is the use of another sort of subgroup tower, one which is not easily observable solely in terms of the action on the original permutation domain. Its very specification subsumes knowledge of the group structure. We construct an SGS with respect to a subgroup chain $G = G_0 \geq G_1 \geq \cdots \geq G_m = 1$ such that each G_i is *normal* in G and the factor groups G_i/G_{i+1} are either products of isomorphic alternating groups or subgroups of products of small primitive groups ("small" in this context means of order $n^{c \log n}$).

Naive divide-and-conquer of the permutation domain provides some normal subgroups of G in the kernels of induced actions on orbits or blocks of imprimitivity; the new machinery comes into play precisely when such decomposition bottoms out. The structure of large primitive groups allows an augmentation of the domain that readmits naive decomposition. This idea is part of the NC-procedure developed for the same problem [BLS87]. However, the sequential algorithm cannot be viewed as the sequential implementation of the parallel one. Even a knowledgeable implementation of the relevant part of parallel ideas would require $O^{\sim}(n^6)$ at best.

The timing analysis depends on the classification of finite simple groups via information on the order of primitive permutation groups whose socle is not the product of alternating groups. We remark, however, that there is an *elementary* version of the algorithm breaking the $O(n^5)$ barrier. Instead of the classification, we may use Babai's bound [Ba] on the order of uniprimitive groups and Pyber's recent bound [Py] on the order of doubly transitive groups to obtain an $O^{\sim}(n^{4.5})$ algorithm. In fact, the elementary algorithm is simpler in the sense that we do not have to detect alternating groups in socles of primitive groups involved in G (unless the primitive group itself is alternating or symmetric in its natural action on blocks of imprimitivity of G). Both Babai's and Pyber's results are within a logarithmic factor (in the exponent) from optimal; the loss in running time is due to the fact that we do not have elementary

estimates for the order of primitive groups with nonalternating-type socle. We sketch the elementary version in section 9.

We mention two further aspects which are important differences from previous methods. Exploiting the normality of subgroups in the new subgroup chain, we first obtain only normal generators, i.e., generators whose normal closure is the given subgroup. Another difference is the novel handling of full symmetric and alternating groups. We formulate the latter result as a separate theorem.

THEOREM 1.2. *From a given list of generators of the symmetric or alternating group, one can construct an SGS with respect to the chain of point stabilizer subgroups in* $O^{\sim}(n^3 + sn^2)$ *time. (The term "construct" refers to the operations of taking products and powers of permutations.) Moreover, there is a Las Vegas algorithm achieving the same goal in* $O^{\sim}(n^2 + sn)$ *expected running time.*

A random algorithm is Las Vegas if it never returns incorrect answers. We require that the SGS is constructed from the given generators via permutation multiplications since we apply this result when the symmetric group is involved in a larger permutation group G and acts on blocks of imprimitivity of G. We can guarantee that a given permutation from the symmetric group belongs to G only if it is constructed by the aforementioned operations.

We remark that the *random subproduct method*, originally developed to prove the random part of Theorem 1.2, was substantially extended by Babai, Cooperman, Finkelstein, Luks, and Seress [BCFLS91], [BCFLS95] to yield an elementary Monte Carlo algorithm for the basic tasks mentioned in Theorem 1.1 which runs in $O(n^3 \log^4 n + sn \log n)$ time. (A Monte Carlo algorithm may return a wrong answer with a fixed but arbitrarily small probability.)

In a sequel to this paper, we shall extend our method to achieve a further order of magnitude improvement in the running time.

THEOREM 1.3 (see [BLS]). *Given a permutation group* G *by a list* S *of generators,* $|S| = s$*, the following problems can be solved in* $O^{\sim}(sn^3)$ *time:*

(a) *All items listed in Theorem* 1.1.

(b) *Finding a composition series of* G.

Let us remark that the length of the input is $\Theta(sn)$ so this is an $O^{\sim}(n^3)$ algorithm as a function of the input length. For the more complicated task of computing a composition series, Theorem 1.3 gives an improvement of *five* orders of magnitude from Luks's original algorithm [Lu87]. This result requires a deeper probe into the structure of primitive groups with different types of socle, in the spirit of the O'Nan–Scott theorem [Sc], [Cam].

Like the method of this paper, the $O^{\sim}(sn^3)$ algorithm examines the primitive groups involved in G and locates the large alternating composition factors. It differs in its handling of the nonalternating part of G and a reduction of the number of "normal generators" for consecutive groups in the normal series. Specifically, the arguments in sections 6 and 7 are improved. A part of these results appeared in [BLS93].

As presented in sections 3–8, our $O^{\sim}(n^4)$ algorithm requires $O^{\sim}(n^3 + sn^2)$ memory. In section 9 we indicate how to decrease the memory requirement to $O^{\sim}(n^2 + sn)$. Also, without loss in time efficiency, the algorithm can output Jerrum's compressed data structure [Je86] for an SGS with respect to the point stabilizer subgroup chain; this requires only $O(n^2)$ space and supports membership testing in $O(n^2)$ time per test.

At this point, our emphasis is on the theoretical improvement realized by our

algorithm. Practical computations often deal with so-called *small-base groups*, i.e., families of groups satisfying $\log|G| < \log^c n$ for some fixed constant c. For small-base group inputs, the traditional algorithms run in $O^{\sim}(n^2)$ time and our method becomes essentially a version of the traditional approach. The attention given to the small-base case is, in part, due to the fact that interesting permutation representations of the nonalternating simple groups tend to have a small base. However, it is also the case that it has often been impractical to deal with large-base groups. Thus, aspects of the new methods should be important in practice where there is a need to deal with groups where $\log|G|$ is, say, proportional to n and when hardware is improved to allow the usage of $\Theta(n^2)$ memory for n in the tens of thousands.

2. Definitions and preliminaries. We refer to any standard text, e.g. [Ha], for basic facts about groups. For permutation group concepts we refer to [Wi] and [Cam]. We mention two sources of information on the classification of finite simple groups [Go], [Car], but no knowledge of these works is required. Cameron [Cam] gives a fine survey of all the consequences of the simple groups classification relevant to our work.

2.1. Group theory. We write $H \leq G$ if H is a subgroup of G and $H \triangleleft G$ if H is a normal subgroup of G.

LEMMA 2.1 (see [Ha, p. 96]). *Let $H \leq G$ and assume S is a set of generators of G and R is a complete set of right coset representatives of G mod H. Then the set*

$$\{\rho\sigma\rho_1^{-1} : \rho, \rho_1 \in R, \sigma \in S, \rho\sigma\rho_1^{-1} \in H\}$$

generates H.

The generators described here are called *Schreier generators* of H; their number is $|S||G:H|$ (these are not necessarily distinct).

For $Q \subset G$, the *normal closure* $\langle Q^G\rangle$ of Q in G is the smallest normal subgroup of G containing Q. More generally, for $K \leq G$, $\langle Q^K\rangle$ is the smallest subgroup of G containing Q and normalized by K. We say Q is a set of *normal generators* for H if $H = \langle Q^G\rangle$. For $\tau, \sigma \in G$, τ^σ denotes the *conjugate* $\sigma^{-1}\tau\sigma$. A group $G \neq 1$ is called *simple* if it has no nontrivial normal subgroups. We call G *semisimple* if it is the direct product of simple groups. If these simple groups are isomorphic then G is *characteristically simple*. A *composition series* of G is any series $1 = G_r \triangleleft \cdots \triangleleft G_1 \triangleleft G_0 = G$ where the quotients G_{i-1}/G_i are simple; these quotients are the *composition factors*. The group G is *solvable* if all composition factors of G are cyclic. We need the following well-known fact (see, e.g., [Sc]).

PROPOSITION 2.2. *Let H be a subgroup of the semisimple group $G = \prod_{i=1}^m T_i$ such that all T_i are simple nonabelian and H projects onto each factor. Then H is direct product of "diagonal" subgroups; more precisely, the T_i can be arranged into blocks of isomorphic groups so that, after a suitable renumbering of the factors,*

$$H = \mathrm{Diag}(T_1 \times \cdots \times T_{k_1}) \times \cdots \times \mathrm{Diag}(T_{k_{r-1}+1} \times \cdots \times T_{k_r}).$$

In other words, having identified the groups in each block, H consists precisely of the elements of the form

$$(\alpha_1, \ldots, \alpha_1), \ldots, (\alpha_r, \ldots, \alpha_r).$$

The *socle* of G is the subgroup generated by all minimal normal subgroups and is denoted by $\mathrm{Soc}(G)$. The socle is semisimple.

The *automorphism group* of G is denoted by $\mathrm{Aut}(G)$. Every element $g \in G$ induces an *inner automorphism* $x \mapsto g^{-1}xg$. The group of inner automorphisms, $\mathrm{Inn}(G)$, is normal in $\mathrm{Aut}(G)$. The factor group $\mathrm{Out}(G) = \mathrm{Aut}(G)/\mathrm{Inn}(G)$ is the *outer automorphism group*. One of the classification-dependent results required by our algorithm analysis is the so-called Schreier conjecture.

THEOREM 2.3 (Schreier conjecture). *The outer automorphism group of a finite simple group is solvable.*

2.2. Permutation groups. The group of all permutations of an n-element set A is denoted $\mathrm{Sym}(A)$, or $\mathrm{Sym}(n)$ if the specific set is unessential. Subgroups of $\mathrm{Sym}(n)$ are the *permutation groups* of *degree* n. The *even* permutations of A form the *alternating group* $\mathrm{Alt}(A)$ (or $\mathrm{Alt}(n)$). We refer to $\mathrm{Sym}(A)$ and $\mathrm{Alt}(A)$ as the *giants*. These two families of groups require special treatment in most algorithms (see sections 5 and 8).

The *support* $\mathrm{supp}(\pi)$ of $\pi \in \mathrm{Sym}(A)$ consists of those elements of A actually displaced by π, i.e., $\{a \in A : a^\pi \neq a\}$. The *degree* of π is $\deg(\pi) = |\mathrm{supp}(\pi)|$.

We say that G *acts on* A if a homomorphism $G \to \mathrm{Sym}(A)$ is given. This action is *faithful* if its kernel is the identity. The *orbit* of $a \in A$ under G is the set of images $\{a^\gamma : \gamma \in G\}$. G is *transitive* on A if there is only one orbit. We say G is t*-transitive* if the action of G induced on the set of ordered t-tuples of distinct elements of A is transitive ($t \leq n$). The maximum such t is the *degree of transitivity* of G. The degree of transitivity of the giants is $\geq n - 2$.

THEOREM 2.4. *If G is 2-transitive and $|G| \geq n^{2+\log n}$ then G is giant.*

This is an immediate consequence of the classification of doubly transitive groups, which is essentially due to Curtis, Kantor, and Seitz [CKS]. Their work is based on detailed knowledge of the finite simple group classification. For the list of doubly transitive groups, see, e.g., [Cam].

Actually, we could use a weaker version of Theorem 2.4, with no loss in the asymptotic analysis of running time. The following result has a strikingly simple, elementary proof.

THEOREM 2.5 (see [Py]). *There exists an explicitly computable constant c such that if G is 2-transitive and $|G| \geq n^{c \log^2 n}$ then G is giant.*

2.3. Orbits, orbitals, blocks, stabilizers. If G acts on A, the orbits of the induced (componentwise) G-action on $A \times A$ are called *orbitals* [Si67]. The *stabilizer* of $x \in A$ is the subgroup $G_x = \{\gamma \in G : x^\gamma = x\}$. If G is transitive on A then there is a bijection between the orbitals of G and the orbits of G_x. For an orbital Θ of G and $x \in A$, the (out)neighbors of x in the (di)graph (A, Θ) form the orbit $\Theta(x) = \{y | (x, y) \in \Theta\}$ of the stabilizer G_x. For $B \subset A$, we use G_B for the *pointwise stabilizer* $\bigcap_{x \in B} G_x$ of B, and $G_{\{B\}}$ for the setwise stabilizer $\{\gamma \in G : B^\gamma = B\}$ of B. If $B \subset A$ is stabilized by G, then we denote by G^B the restriction of G to B, so that $G^B \leq \mathrm{Sym}(B)$. Then, $G(B) = G^B_{\{B\}}$ denotes the image of the action of the setwise stabilizer of B on B.

If G is transitive on A and $G_x = 1$ for some (any) $x \in A$, then G is said to be *regular*. If G is transitive and $D \subseteq A$, D is called a *block* (for G) if for all $\gamma \in G$, either $D^\gamma = D$ or $D^\gamma \cap D = \emptyset$, and G is called *primitive* if no nontrivial blocks exist. (Trivial blocks have 0, 1, or $|A|$ elements.) If D is a block then the set of images of D is called a *block system* and an action of G is induced on the block system. The block system is *minimal* if that action is primitive.

For section 4, we need the following elementary results on the structure of primitive groups. They all follow from the O'Nan–Scott theorem [Sc] (cf. [Cam], [Lu87]).

THEOREM 2.6. *Let $G \leq \mathrm{Sym}(A)$ be primitive. If $\mathrm{Soc}(G)$ is abelian then $n = p^d$ for some prime p, A can be identified with the d-space over $GF(p)$ and (via this identification) $G \leq \mathrm{AGL}(d,p)$, the group of affine transformations of A, and $\mathrm{Soc}(G) \cong \mathbf{Z}_p^d$ is the group of translations of A.*

THEOREM 2.7. *Let $G \leq \mathrm{Sym}(A)$ be primitive. Then*

$$\mathrm{Soc}(G) = T_1 \times \cdots \times T_d$$

where the T_i are isomorphic simple groups. If $\mathrm{Soc}(G)$ is nonabelian then G contains a normal subgroup N such that

(a) $\mathrm{Soc}(G) \leq N \leq \mathrm{Aut}(T_1) \times \cdots \times \mathrm{Aut}(T_d)$;

(b) *G/N is a subgroup of S_d;*

(c) $n \geq 5^d$.

In the particular case that the isomorphic T_i are alternating groups, we say that G is of *alternating type*.

THEOREM 2.8. *Let $G \leq \mathrm{Sym}(A)$ be primitive. If G has more than one minimal normal subgroup then G has precisely two minimal normal subgroups, each of order $|A|$.*

2.4. Primitive groups of Cameron type. A remarkable class of primitive groups of alternating type is obtained as follows.

First we define a class of imprimitive groups. Let B be a set of k elements, $k \geq 5$, and $1 \leq s < k/2$. Let $C = rB = B_1 \dot\cup \cdots \dot\cup B_r$ denote the disjoint union of r copies of B. An *s-transversal* of C is a subset $X \subset C$ such that $|X \cap B_i| = s$ for $i = 1, \ldots, r$. Let A denote the set of s-transversals and let $n = |A| = \binom{k}{s}^r$. The *wreath product* $W(B,r) = \mathrm{Sym}(B) \wr S_r \leq \mathrm{Sym}(C)$ consists of all permutations of C that respect the partition $\{B_i\}$. Clearly,

$$\mathrm{Soc}(W(B,r)) = \mathrm{Alt}(B_1) \times \cdots \times \mathrm{Alt}(B_r).$$

Now let $W(B,r) \geq G \geq \mathrm{Soc}(W(B,r))$ and assume G acts transitively on the set of blocks $\{B_i\}$. Under these conditions, the action of G on A is *primitive* (and alternating type, since $\mathrm{Soc}(G) = \mathrm{Soc}(W(B,r))$). We say that the primitive groups obtained this way are of *Cameron type*.

THEOREM 2.9 (see [Li]). *If G is a primitive group of degree n and order $> n^{9 \log n}$ then G is of Cameron type.*

This is the third consequence of the simple groups classification that we require. The name "Cameron type" acknowledges the first version of Theorem 2.9 by Cameron [Cam], who formulated the lower bound as $> n^{c \log n}$, without explicit determination of the constant $c = 9$. For large n, c approaches 1. We remark that the actual value of c plays no role in the algorithms; their analysis depends only on the existence of c.

2.5. Cameron schemes. For application in section 4, we introduce a combinatorial structure associated with the action of $W(B,r)$ on A. Let A, B, C be as above. For an s-transversal $X \in A$, let $X_i = X \cap B_i$. For $X, Y \in A$, let $d_i = |X_i \cap Y_i|$ and let $f_1 \leq f_2 \leq \cdots \leq f_r$ be the sorted sequence $\{d_i\}$. We call $(f_1, \ldots, f_r)$ the *intersection pattern* of X and Y. Let us partition $A \times A$ according to intersection patterns: $A \times A = R_0 \cup \cdots \cup R_N$. We call the system $C(n,k,s,r) = (A; R_0, \ldots, R_N)$ the *Cameron scheme* with parameters (n,k,s,r). This is a particular *association scheme*

[Bos], [Del], [MS]; it includes the Hamming schemes ($s = 1$) and the Johnson schemes ($r = 1$) as particular cases. The scheme can be thought of as a coloring of the edges of the complete graph on n vertices (including self-loops); we refer to the R_i as *color classes*.

It is clear that each group of Cameron type acts on a Cameron scheme. In fact, the color classes are precisely the orbitals of the action of $W(B, r)$ on A. It may, however, happen that the color classes split under the action of a Cameron-type group $G \le W(B, r)$. In a key subroutine, NATURAL_ACTION, we recover the imprimitive action of G on $C = rB$ using the orbital structure of the primitive G-action on A, thereby reducing the Cameron-type groups to imprimitive groups with a block system of $r \le \log n / \log 5$ blocks, with giants acting on each block.

Some elementary observations about the orbital structure will be useful in this computation. Let Σ_i be the color class corresponding to the intersection pattern $(s - i, s, \dots, s)$ and Φ to $(0, 0, \dots, 0)$.

LEMMA 2.10. *Let G be a Cameron-type group acting on the points A of a Cameron scheme $C(n, k, s, r)$ and suppose $k \ge 2rs^2$. Then the following hold.*

(a) *Σ_1 is the second smallest orbital of G.*

(b) *Φ is the largest orbital of G.*

Proof. We note first that Σ_i $(0 \le i \le s)$ and Φ are orbitals of G; i.e., they do not split. For Φ this follows from the fact that $G \ge \mathrm{Alt}(k)^r$. For Σ_i we need in addition that the stabilizer of any $a \in A$ acts transitively on the set of blocks $\{B_i\}$.

Proof of (a): Fix $x \in A$ and consider an orbital Θ. We have to prove that $|\Sigma_1(x)| < |\Theta(x)|$ for any Θ other than Σ_1 and the diagonal Σ_0 (the diagonal is the *smallest* orbital). Observe, since $k \ge 2s^2$, that $\binom{k-s}{s} \ge s(k - s)$ with strict inequality when $s > 1$.

For $s \ge i > 1$,

$$|\Sigma_i(x)| = r\binom{s}{s-i}\binom{k-s}{i} > rs(k-s) = |\Sigma_1(x)|.$$

Assume now that Θ is contained in the color class with intersection pattern $(i_1, i_2, \dots)$ where $i_1 \le i_2 < s$; let $(x, y) \in \Theta$. Just counting the images of y under the stabilizer of x in $\mathrm{Alt}(k)^r$ we obtain

$$|\Theta(x)| \ge \binom{s}{i_1}\binom{k-s}{s-i_1}\binom{s}{i_2}\binom{k-s}{s-i_2}$$

$$\ge s^2(k-s)^2 > rs(k-s),$$

the final inequality using $k \ge 2r$.

Proof of (b): We have to prove that Φ is the largest color class in the Cameron scheme. (Note that G plays no role here.)

We use the fact that, for $1 \le i \le s$,

$$r\binom{k-s}{s-i}\binom{s}{i} < \binom{k-s}{s}.$$

To see this, note that $k \geq \max\{2rs^2, 2s+1\}$ implies $k - 2s + 1 > rs^2$, so that

$$\frac{\binom{k-s}{s}}{\binom{k-s}{s-i}} = \prod_{j=0}^{i-1} \frac{k-2s+i-j}{s-j} > \prod_{j=0}^{i-1} \frac{rs^2}{s-j} \geq rs^i \geq r\binom{s}{i}.$$

Now let the color class Θ have intersection pattern $(0^{r_0}, \ldots, s^{r_s})$. (The exponents denote multiplicities.) Then

$$|\Theta(x)| = \binom{r}{r_0, r_1, \ldots, r_s} \prod_{i=0}^{s} \binom{k-s}{s-i}^{r_i} \binom{s}{i}^{r_i}$$

$$< \binom{r}{r_0, r_1, \ldots, r_s} \binom{k-s}{s}^{r} \frac{1}{r^{r-r_0}} < \binom{k-s}{s}^{r} = |\Phi(x)|. \qquad \square$$

2.6. Strong generators. In our algorithms, permutation groups are input and output via sets of generators. A standard tool for permutation group computation is an SGS [Si70]. An SGS with respect to the subgroup chain $G = G_0 \geq G_1 \geq \cdots \geq G_m = 1$ is a set $T \subset G$ such that $T \cap G_i$ generates G_i for all i.

Let C_i be a set of (right) coset representatives for G_{i-1} mod G_i, $i = 1, 2, \ldots, m$. Then any $\alpha \in G$ has a unique factorization $\alpha = \rho_m \cdots \rho_2 \rho_1$ with $\rho_i \in C_i$. An SGS T is computationally effective if, for any $\alpha \in G_{i-1}$, there are fast procedures for determining the coset of G_i to which α belongs and constructing a representative for this coset from T.

We construct an SGS with respect to a subgroup chain $G = G_0 \geq G_1 \geq \cdots \geq G_m = 1$ such that each G_i is normal in G and the factor groups G_{i-1}/G_i are either subgroups of direct products of small primitive groups ("small levels") or direct products of alternating groups ("alternating levels"). To achieve the effectiveness mentioned above, we, in fact, construct an SGS with respect to a refinement $G = H_0 \geq H_1 \geq \cdots \geq H_{m'} = 1$ of the subgroup chain $G = G_0 \geq G_1 \geq \cdots \geq G_m = 1$. Namely, we construct a permutation representation for each G_{i-1} with kernel G_i. The refinement between G_{i-1} and G_i is a pointwise stabilizer chain in this representation. The advantage of a pointwise stabilizer chain is that it is easy to recognize the coset to which a given permutation belongs: given $\alpha \in H$, its coset $H_x\alpha$ is determined by x^α.

If a pointwise stabilizer chain is long, it requires too much time and storage to store all coset representatives at each level. Hence, in alternating groups, we use the following Jerrum-style [Je86] compact SGS. Suppose that $K \cong \mathrm{Alt}(m)$ and K acts naturally on a set $B = \{x_1, \ldots, x_m\}$ with K_i the pointwise stabilizer of $\{x_1, \ldots, x_i\}$. Let the set T consist of the permutations $\pi_1, \pi_2, \ldots, \pi_{m-1}$ satisfying the following properties. For all $1 \leq k \leq m-2$, π_k fixes pointwise $x_1, \ldots, x_{k-1}$ and π_{m-1} fixes pointwise $x_1, \ldots, x_{m-3}$. Moreover, $x_k^{\pi_k} = x_{k+1}$ for $k = 1, 2, \ldots, m-2$ and $x_{m-2}^{\pi_{m-1}} = x_m$. Suppose we store just the products $\mu_i = \pi_1\pi_2\cdots\pi_i$ for $i \leq m-2$ and $\mu_{m-1} = \pi_1\pi_2\cdots\pi_{m-3}\pi_{m-1}$. Then $\{\mu_{i-1}^{-1}\mu_j : i-1 \leq j \leq m-1\}$ is a complete set of coset representatives for K_i in K_{i-1}. Thus, any coset representative within the chain can be obtained with one multiplication. (We use the term *multiplication* for the evaluation of $\alpha^{-1}\beta$ as well as $\alpha\beta$; clearly the timings are the same.)

It is useful to observe that an SGS for a factor group G/N, lifted to G and appended to an SGS for N, gives an SGS for G. With an abuse of language, we call a subset $C \subset G$ a set of strong generators of G/N if C is a lifting of such a set. Suppose $\alpha \in G$ is factored according to a fixed SGS of G/N, that is, $\bar{\alpha} = \bar{\rho}_l \cdots \bar{\rho}_2 \bar{\rho}_1$, where the bar signifies the image mod N. Then $\nu = \alpha(\rho_l \cdots \rho_1)^{-1} \in N$ and we call ν the *siftee* of α into N. The following notion plays an important role in reducing the number of generators we use for N. If $C \subset G$ is an SGS for G/N and $S^* \subset G$ is a set of generators for G/N such that $C \subset \langle S^* \rangle$ (in G, not only in G/N), then we say that S^* is *compatible* with C.

2.7. Sims's algorithm. Sims's algorithm for constructing strong generators has been formulated for the case when G_i is the stabilizer of the first i points of the permutation domain. An efficient version of Sims's method has been analyzed by Knuth [Kn]. In this subsection, we describe a slight extension of the latter version.

We consider the action of $G = \langle Q \rangle \leq \mathrm{Sym}(A)$ on the set $C = \{1, 2, \ldots, m\}$. Let G_i be the pointwise stabilizer of $\{1, 2, \ldots, i\}$,

$$G = G_0 \geq G_1 \geq \cdots \geq G_m = N,$$

where N is the kernel of the G-action on C. Our objective is to find an SGS of G/N.

During the procedure, we maintain lists T_i, $i = -1, 0, \ldots, m-1$ and R_i, $i = 0, 1, \ldots, m$, where R_i is a not-necessarily-complete list of right coset representatives of G_{i-1} mod G_i; and $T_i \subseteq G_i$ such that $\langle T_i \rangle \supseteq \bigcup_{j \geq i+1} R_j$. The lists $T_{-1} := Q$ and $R_0 := \{1\}$ do not change during the procedure; all other lists may sometimes be augmented. Each time T_i is augmented, the group $\langle T_i \rangle^C$ increases.

We employ the following SIFT routine which attempts to factor $\pi \in G_k$ (k is part of the input) over the current partial coset lists. If it does not succeed then it inserts a new coset representative in the appropriate R_{j+1}, updates the T_i, $k+1 \leq i \leq j$, and sets $k := j$. In any case, at the conclusion of SIFT, $\pi \in N R_m R_{m-1} \cdots R_{k+1}$.

procedure SIFT$(\pi, C, k, \{T_i\}, \{R_i\})$
Initialize: $\sigma := \pi$, $j := k$.
    while $j \leq m-1$ **and** $\sigma \neq 1$ **do**
        if $\sigma \in G_{j+1}\alpha$ for some $\alpha \in R_{j+1}$
            then set $\sigma := \sigma\alpha^{-1}$ and $j := j+1$
            else
                begin
                    add σ to R_{j+1};
                    add σ to T_l, $l = k+1, \ldots, j$;
                    $k := j$;
                end ;
end (SIFT).

The main procedure is the following.

procedure PERMREP(Q, C)
INPUT: (Q, C) as above.
OUTPUT: $\{T_i\}, \{R_i\}$.
Initialize: $k := -1$, $T_{-1} := Q$,
        $T_i := \emptyset$ for $0 \leq i \leq m-1$, $R_i := \{1\}$, for $0 \leq i \leq m$.
    while $k \geq -1$ **do**
        begin

```
        while R_{k+1} × T_k not exhausted do
            begin
                select next (ρ, τ) in R_{k+1} × T_k;
                SIFT(ρτ, C, k, {T_i}, {R_i});
            end ;
        k := k − 1
        end
end (PERMREP).
```

Note that the intention is to put the elements of each $R_{k+1} \times T_k$ in a queue as such elements are created (by augmentation of R_{k+1} and/or T_k), ensuring that each $(\rho, \tau) \in R_{k+1} \times T_k$ is selected exactly once in the lifetime of the procedure.

The following proposition is just a reformulation of Sims's basic observations.

PROPOSITION 2.11. *When procedure PERMREP(Q, C) terminates, R_i is a complete set of coset representatives for G_{i-1} mod G_i and $\langle T_i \rangle^C = G_i^C$ for $0 \le i \le m$.*

Proof. Recall that N denotes the kernel of the G-action on C. As a result of having sifted $R_i T_{i-1}$, we know $R_i T_{i-1} \subseteq N R_m R_{m-1} \cdots R_i$ for $0 \le i \le m$. We have also maintained the properties $T_i \subseteq G_i$, $\bigcup_{j \ge i+1} R_j \subseteq \langle T_i \rangle$, and the elements of R_i represent distinct cosets mod G_i.

Since $Q = R_0 T_{-1} \subseteq N R_m \cdots R_1 \subseteq N\langle T_0 \rangle$, $G = N\langle T_0 \rangle$. Suppose, for any i that $G_i = N\langle T_i \rangle$; then $R_{i+1} T_i \subseteq N R_m R_{m-1} \cdots R_{i+1} \subseteq N\langle T_{i+1} \rangle R_{i+1}$, whence $G_i \subseteq N R_{i+1} \langle T_i \rangle \subseteq N\langle T_{i+1} \rangle R_{i+1}$. It follows that $G_i = N\langle T_i \rangle = N G_{i+1} R_{i+1}$ and $G_{i+1} = N\langle T_{i+1} \rangle$. □

We use the following easy facts about PERMREP(Q, C). We set $n = |A|$ and assume $n \ge m = |C|$. Therefore, the cost of each group operation is $O(n)$. Let $t = \max\{|G_{i-1}^C : G_i^C| : 1 \le i \le m\}$; note that $t \le m$. Also, $\log |G^C|$ is an upper bound on the length of subgroup chains in G^C so there are $\le \log |G^C|$ indices i such that $G_i \ne G_{i+1}$. In particular, the cost of each sift is $O(n \log |G^C|)$ and each T_i is increased $\le \log |G^C|$ times. From this, we obtain the following estimates.

THEOREM 2.12. (a) *Let $|Q| = q$. The running time of PERMREP(Q, C) is $O(n \log |G^C|(q + t \log^2 |G^C|))$; in particular, if $|G^C| \le \exp(\log^c n) = \exp(O^\sim(1))$ then the running time is $O^\sim(n(q + t))$.*

(b) *At any moment during the execution of the algorithm, $|T_{k-1}| \le 1 + \sum_{j=k}^m \log |R_j|$.*

Remark 2.13. The $O(n^5)$ bottleneck that is inherent to all versions of Sims's method [Si70], [FHL], [Je86], [Kn] can be appreciated in the context of PERMREP (take $C = A$ and $N = 1$). These methods rely on the construction of generators for the groups in the pointwise stabilizer chain, using Schreier's construction of subgroup generators. (In PERMREP, Schreier generators enter in the sifting of $R_{k+1} T_k$, since the sift of $\rho\tau$ begins with finding $\rho_1 \in R_{k+1}$ such that $\rho\tau\rho_1^{-1} \in G_{k+1}$.) In general, $|R_{k+1}|$, $|T_k|$ and the number of groups in the chain can each be $\Omega(n)$ so there may be $\Omega(n^3)$ elements to sift and a sift may cost $\Omega(n^2)$. In fact, Knuth discusses a class of groups in which the *average* behavior of such methods is $\Theta(n^5)$.

As in [FHL], a slight modification of PERMREP provides normal closures. The addition to the previous procedure is that we have to add conjugates of generators to the generating set until we get a group closed for conjugation. We again consider group actions.

The situation is the following: $G = \langle S \rangle \le \mathrm{Sym}(A)$ and $\langle Q \rangle \le \mathrm{Sym}(A)$ act on C. The output consists of sets of coset representatives $\{R_i\}$ and sets of generators of the stabilizer chain over C for $H := \langle Q^G \rangle$. For sets of permutations T and S, T^S denotes

the set of conjugates $\{\tau^\sigma : \tau \in T, \sigma \in S\}$.

procedure NORMCL(Q, C, S)
INPUT: (Q, C, S) as above.
OUTPUT: $\{T_i\}, \{R_i\}$.
    PERMREP(Q, C);
    $T^* := \emptyset$;
    repeat
        $k := -1$, $T_{-1} := (T_0 \setminus T^*)^S$, $T^* := T_0$;
        while $k \geq -1$ **do**
            begin
                while $R_{k+1} \times T_k$ not exhausted **do**
                    begin
                        select next $(\rho, \tau) \in R_{k+1} \times T_k$;
                        SIFT$(\rho\tau, C, k, \{T_i\}, \{R_i\})$;
                    end ;
                $k := k - 1$
            end
    until $T_0 = T^*$
end (NORMCL).

The proof of correctness and the timing of this algorithm is similar to that of PERMREP (with H playing the role of G in the estimates). Let $t = \max\{|H_{i-1}^C : H_i^C| : 1 \leq i \leq m\}$.

PROPOSITION 2.14. *When procedure NORMCL(Q, C, S) terminates, the R_i form complete sets of coset representatives for H, and $\langle T_i \rangle^C = H_i^C$.*

THEOREM 2.15. *Let $s = |S|$, $q = |Q|$. The running time of NORMCL(Q, C, S) is $O(n \log |H^C|(q + s \log |H^C| + t \log^2 |H^C|))$; in particular, if $|H^C| \leq \exp(\log^c n) = \exp(O^\sim(1))$ then the running time is $O^\sim(n(q + s + t))$.*

2.8. Structure forest, structure domain. It is natural in dealing with permutation groups, whether theoretically or in computational settings, to use the orbit structure in a problem decomposition. Further combinatorial divide-and-conquer is available in the imprimitivity structure of transitive groups. For computational purposes, it is convenient to provide an extension of the permutation domain that both reflects and guides the flow of control in such procedures. Specifically, a *structure forest* (SF) for a permutation group $G \leq \mathrm{Sym}(A)$ is a forest of rooted trees on which G acts as automorphisms fixing the roots, such that the leaves form the permutation domain A, and, denoting by $G(v)$ the permutation group induced on the children of node v by G_v (the stabilizer of v), each $G(v)$ is primitive. Thus, in particular, there is exactly one tree per orbit in A, and it is not possible to insert intermediate levels in that tree, with nontrivial branching, and remain consistent with the G action on the tree.

To reflect the flow of control in our procedure (e.g., treating orbits sequentially) we suppose that the trees of the SF are stacked vertically and enumerate the resulting "levels." Hence, the root of the first tree comprises level 0, its leaves comprise level h, where h is the height of this tree, the root of the second tree comprises level $h + 1$, etc.

The divide-and-conquer offered by the SF alone does not suffice for our methods. To achieve a finer decomposition, we need to delve into the primitive groups themselves; specifically, for "large" groups, we use the forced relations between the nature

of the socle and that of the permutation domain.

The first and principal stage creates an *extended structure forest* (ESF). For this, the SF is augmented at nodes v where $G(v)$ is a "large" group, i.e., of order $> \exp(\log^c n)$. At such places, we are assured that $G(v)$ is, in fact, a Cameron-type group with $\mathrm{Soc}(G(v)) \cong \mathrm{Alt}(k)^r$. Such $G(v)$ has a natural imprimitive representation on a set B of size kr, and we can build a structure forest (in fact, a tree) $T(v)$ on B for $G(v)$. Our algorithm constructs the trees $T(v)$ so that the leaves of $T(v)$ correspond to certain subsets of the children of $G(v)$. In particular, we need only do the work of constructing $T(v)$ at one node v at each level of the SF, using the action of G to copy the trees to other nodes at the same level. As a result, the permutation action of each element of G naturally extends to the ESF. We consider the trees $T(v)$ appended to the SF to be placed *entirely between levels of the initial forest*. Having so situated the $T(v)$, we delete the edges between v and its children in the original forest. Thus, edges, where they exist, in the ESF only traverse consecutive levels. It is important to note, however, that $G(v)$ acts faithfully on the leaves of $T(v)$, so that the subgroup of G that fixes all the leaves at this level also fixes all the nodes at the level of the children of v in the SF.

We now continue to use $G(v)$ to denote the (primitive) permutation group induced by G_v on the set of children of the node v of the ESF. (In context, it is clear which $G(v)$ is intended when we specify the ambient graph.) Thus, in the ESF, $G(v)$ is either a "small" group (of order $< \exp(\log^c n)$) or a giant. Furthermore, the groups at a given level are isomorphic, in fact, conjugate under the action of G; accordingly, we can talk about *small group levels* and *giant levels* in the ESF.

A second refinement is used to restrict the giant levels to be alternating. Consider a node v of the ESF where $G(v)$ is a full symmetric group $\mathrm{Sym}(C(v))$ on the children $C(v)$ of v. At each such node, we append a small tree consisting of the root v and two leaves, say v_L and v_R (for "left" and "right"), which are inserted at a new level between v and $C(v)$. Again, we sever the links from v to $C(v)$, but we now connect both v_L and v_R to all points in $C(v)$. We need to extend the action of G to the new intermediate level. This may be done by fixing any orderings of the $C(v)$, relative to which the actions of $\gamma \in G$ can be viewed as inducing even or odd permutations; if γ induces an "even" mapping of $C(v)$ to $C(w)$ then $v_L^\gamma = w_L$ and $v_R^\gamma = w_R$, else $v_L^\gamma = w_R$ and $v_R^\gamma = w_L$. We call the resulting structure D a *structure domain* (SD) for G.

For a node $v \in D$, we continue to denote the children of v, that is, the neighbors at next level by $C(v)$ and the action of G_v on $C(v)$ by $G(v)$.

We summarize some important properties of this structure. A structure domain for $G \leq \mathrm{Sym}(A)$ is a graph $D = (V, E)$ such that the following hold.

(i) $A \subseteq V$ and $|V| = O(n)$, where $n = |A|$.

(ii) The action of G extends to $\mathrm{Aut}(D)$.

(iii) The orbits of G in V, called "levels," are ordered, $L_0, L_1, \ldots, L_m$, and $E \subseteq \bigcup_{i=0}^{m-1}(L_i \times L_{i+1})$.

(iv) If $E \cap (L_i \times L_{i+1}) = \emptyset$, then $G_{L_i} \leq G_{L_{i+1}}$.

(v) If $E \cap (L_i \times L_{i+1}) \neq \emptyset$, then, letting $G(v)$ denote the action of G_v on the neighbors $C(v)$ in L_{i+1} of $v \in L_i$, it follows that $G(v)$ is either a "small" group or $\mathrm{Alt}(C(v))$.

Let $G_0 = G$ and, for $i \geq 1$, let G_i be the kernel of the action of G_{i-1} on L_i. Then the normal series

$$G = G_0 \geq G_1 \geq \cdots \geq G_m = 1$$

is the chain forecast in the introduction and in section 2.6. Instances of (iv) suggest that the chain is not strictly decreasing (and one can have equality of successive groups even when the induced bipartite graph is nontrivial), but it is convenient to allow this occasional duplication. Note, however, that (v) implies, when $G_{i+1} < G_i$, that G_i/G_{i+1} is either a product of isomorphic alternating groups or a subgroup of the product of isomorphic small primitive groups. The fact that, in the former case, G_{i-1}/G_i is actually isomorphic to a product of alternating groups (not only a subgroup) follows from Proposition 2.2.

Informally, we say the SD consists of *small group levels* and *alternating levels*.

3. Organization of the algorithm. In this section, we outline our main algorithm. Suppose that $G = \langle S \rangle \leq \mathrm{Sym}(A)$ is given, $|A| = n$. We construct a chain of normal subgroups $G = G_0 \geq G_1 \geq \cdots \geq G_m = 1$ and, for each $1 \leq i \leq m$, a permutation representation of G_{i-1} on a set L_i such that

(i) G_i is the kernel of the action of G_{i-1} on L_i;

(ii) G_{i-1}/G_i is either a subgroup of a direct product of small primitive groups ("small" in this context means of order $< \exp(9 \log^2 n \log\log n)$) or $G_{i-1}/G_i \cong \mathrm{Alt}(k)^r$ for some k, r.

The normal subgroup G_i is defined to be the pointwise stabilizer of the first i levels in a structure domain (see section 2.8). However, we have to *construct* generators for the G_i. We do this successively for $i = 0, 1, \ldots, m-1$. We construct an SGS T_i for G/G_i and *normal generators* N_i for G_i, i.e., group elements whose normal closure (in G) is G_i.

Suppose we have constructed T_{i-1} and N_{i-1}. We start to take the normal closure of N_{i-1} in G until the known part of the normal closure generates G_{i-1}/G_i. We confirm this by examining the action of G_{i-1} on L_i. Then we obtain T_i by appending an SGS for G_{i-1}/G_i to T_{i-1}. For this, if G_{i-1}/G_i corresponds to a small group level then we add complete sets of coset representatives from the point stabilizer chain on L_i to T_{i-1} (we do ensure that the total number of saved coset representatives in the entire subgroup chain is only $O(n \log^2 n)$); if G_{i-1}/G_i is the product of alternating groups then we add Jerrum-style compact SGS (cf. section 2.6) for each of these alternating groups to T_{i-1}. We also obtain a presentation for G_{i-1}/G_i, which, along with presentations for earlier quotients, facilitates a construction of normal generators N_i for G_i. Thus we proceed to the next value of i.

We emphasize that generators for G_i (not only normal generators) are available only when the entire algorithm is finished.

During the algorithm, we work with various permutation representations of subgroups of G. If a procedure performs group operations, we may either need the result in the current representation only (*local* operation) or in the original representation as well (*global* operation). An example of a purely local operation is the determination of whether the stabilizer of a node v in the SF acts as a Cameron-type group on its children. To this end, it is enough to perform group operations in $G(v)$ and the action of G_v on other nodes of the SF is irrelevant. All operations not explicitly labelled "local" are understood to be global. Since the sum of sizes of all the induced permutation representations remains $O(n)$, the cost of elementary group operations remains $O(n)$.

Main Algorithm.
INPUT: a set S of generators for $G \leq \mathrm{Sym}(A)$, $|S| = s$.
Step 1. Construct a structure forest and choose a representative v in each orbit of the SF. For all such v, construct Schreier generators for G_v.

Step 2. For these representatives, use NATURAL_ACTION to decide whether $G(v)$ is a "large group" and, if so, construct new action and corresponding structure tree $T(v)$.
Step 3. Append a copy of $T(v)$ to all nodes in the orbit v^G, deleting the connections of v to its children in the SF, thus obtaining an ESF. Extend the G-action of generators to the ESF. Inserting new levels at giant symmetric nodes, obtain the SD. Henceforth, compute the effect of any global operation on the entire SD. Compute the node stabilizers G_w as in Step 1 for representatives of G-orbits of the SD.
Step 4. For each node v representing an alternating level in the SD, construct an SGS for $G(v)$.

Step 5. **for** $i := 1$ to m **do**
construct SGS for G_{i-1}/G_i
store a *compatible* generating set S_{i-1} of size $O^\sim(|L_i|)$ for G_{i-1}/G_i
(* cf. section 2.6 *)
construct normal generators for G_i
end (MAIN ALGORITHM).

LEMMA 3.1. (a) *A structure forest can be computed in* $O(sn^2)$ *time.* (b) *Generators of* G_v *for representatives of the* G*-orbits of the SF can be constructed in* $O(sn^2)$ *time.*

Proof. (a) According to Atkinson [At], a structure forest can be computed as efficiently as orbits and minimal blocks of imprimitivity, i.e., in $O(sn^2)$ time.

(b) The action of the group generators on the orbit v^G of a node v in the SF naturally defines a graph on v^G. Choosing a spanning tree in this graph and computing the product of generators along the paths from v in this tree, group elements which carry v to the other nodes of its orbit can be computed in $O(|v^G|n + |v^G|s)$ time. We obtain generators for G_v (and, at the same time, for $G(v)$) via Lemma 2.1; thus G_v is generated by $O(|v^G|s)$ elements and the cost of computing each is $O(n)$. The result follows since the sum of the $|v^G|$ over orbit representatives v is the number of nodes in the SF. □

Steps 2 and 3 will be analyzed in section 4. We present a novel method for constructing an SGS for the giants in section 5. Section 6 relates group presentations (in terms of generators and relations) to the construction of normal generators. By the results of section 4, the factor groups G_{i-1}/G_i in Step 5 are either subgroups of products of "small" groups or products of alternating groups. We handle the first case in section 7, utilizing NORMCL (cf. section 2.7). For the second case, we give an efficient implementation of Luks's "noncommutative linear algebra" [Lu86] in section 8. Finally, in section 9, we present a version of the algorithm with decreased memory requirement and wrap up the proof of Theorem 1.1.

4. Reducing large to giant. The purpose of this section is to classify primitive groups as "large" and "small." Large groups turn out to be groups of Cameron type, and we construct their "natural" (often imprimitive) action with giants acting on each block and a small group permuting the blocks. Thereby most algorithmic problems are reduced to consideration of giants and small groups.

Our objective is achieved by the subroutine NATURAL_ACTION. This procedure is a slight refinement of the one under the same name in [BLS87]. The procedure involves a global variable n, the degree of the permutation group which is the input of the full algorithm. However, we execute group multiplications only on the set where the group under the current investigation acts primitively (local operations, cf.

section 3).

First, we describe a simple procedure to test whether or not a permutation group is a giant.

procedure TEST_GIANT(G)
INPUT: a 2-transitive group $G = \langle Q \rangle \leq \mathrm{Sym}(C)$, $|C| = m$.
Begin executing PERMREP(Q, C)
if $|\{i : |R_i| \neq 1\}| \geq 2\log m + \log^2 m$ ($*$ we use the notation of section 2.7. $*$)
then stop PERMREP(Q, C); **output** "giant" and **halt**
else output "small group" and **halt**
end (TEST_GIANT).

When reading the following pseudocode, it is useful to review the notation of sections 2.4 and 2.5 and keep in mind that NATURAL_ACTION was designed to handle Cameron-type groups, when $m = \binom{k}{s}^r$ and the underlying set A corresponds to s-transversals in rB for some set B of size k. In that scenario, Γ and Δ correspond to the sets of pairs of s-transversals with intersection pattern $(s-1, s, \ldots, s)$ and $(0, 0, \ldots, 0)$, respectively. The primary aim of the procedure is to construct a subset of A corresponding to the s-transversals containing a fixed point in rB. Such a set is constructed as $C(x, y)$ below, where x, y are s-transversals with intersection pattern $(s-1, s, \ldots, s)$ and $C(x, y)$ consists of all s-transversals containing the unique point in x that is not covered by y. The subset $C(x, y)$ has kr distinct images under G, corresponding to the points of rB. We compute this set D of images in two phases, first constructing in $D(x)$ only the rs images corresponding to the points of x. This and other checks on the sizes of newly constructed objects allow early termination in the case when G is not a large group.

procedure NATURAL_ACTION(Q)
INPUT: a primitive group $G = \langle Q \rangle \leq \mathrm{Sym}(A)$, where $m := |A| \leq n$.
Step 1. **if** $m \leq 3\log^2 n$
then output "small group" and **halt**
Step 2. **if** G is 2-transitive
then TEST_GIANT(G); ($*$ procedure will halt there $*$)
Step 3. Compute the orbitals (G-orbits on $A \times A$);
$\Gamma :=$ the second smallest orbital;
($*$ The smallest orbital is the diagonal. $*$)
$\Delta : D$ the largest orbital.
Fix $x \in A$;
if $|\Gamma(x)| > 2\sqrt{m}\log m$
then output "small group" and **halt**
For each $w \in A$ construct $\alpha(w) \in G$ such that $x^{\alpha(w)} = w$.
Construct Schreier generators for G_x.
Fix $y \in \Gamma(x)$. For each $y' \in \Gamma(x)$ compute some $\beta(y') \in G_x$ such that $y^{\beta(y')} = y'$.
Compute the sets

$$B(x, y) = \Delta(y) - \Delta(x);$$
$$C(x, y) = A - \bigcup_{z \in B(x,y)} \Delta(z).$$

Let $D(x) = \{C(x, y)^{\beta(y')} : y' \in \Gamma(x)\}$.
if $|D(x)| > \log m$
then output "small group" and **halt**

Let $D = \bigcup_{w \in A} D(x)^{\alpha(w)}$.
if $|D| > 2\sqrt{m} \log m$
then output "small group" and **halt**
Step 4. Consider G-action on D. ($*$ This action exists and it is transitive. $*$)
Select a system $\{E_1, \ldots, E_t\}$ of minimal-size (but nonsingleton) imprimitivity blocks
($* \bigcup_i E_i = D *$).
if $q := |E_i| > 4 \log\ n$ **and** $G(E_1)$:=the stabilizer of E_1 restricted to E_1 is 2-transitive
and TEST_GIANT($G(E_1)$) returns message "giant"
then output ("large group, faithfully acting on D";
the G-action on D;
a structure tree for the G-action on D)
else output "small group"
halt
end (NATURAL_ACTION).

We say that G *fails the large groups test* if output is "small group." Otherwise G is said to *pass the large groups test*.

4.1. Correctness of the subroutine NATURAL_ACTION.

LEMMA 4.1. *If TEST_GIANT(G) outputs "giant" then G is a giant. If the output is "small group" then $|G| < m^{2+\log m}$.*

Proof. It is proved by Theorem 2.4. □

The following result justifies the term "small groups" and provides additional information about large groups.

THEOREM 4.2. (1) *If NATURAL_ACTION outputs "giant" then G is a giant.*

(2) *If NATURAL_ACTION outputs "large group" then G acts faithfully on D and the stabilizer of each block E_i restricted to E_i contains* $\mathrm{Alt}(E_i)$.

(3) *If NATURAL_ACTION outputs "small group" then*

$$|G| < \exp(9 \log^2 n\ \log\log n).$$

Statement (1) is obviously correct. For (2) we need a lemma.

LEMMA 4.3. *For $p \neq r$ primes, the order of a Sylow r-subgroup of the linear group* $\mathrm{GL}(d, p)$ *is less than p^{2d}.*

Proof. This result is implicit in [Lu82, Lemma 3.6]. □

COROLLARY 4.4. *For $q \geq 4d \log p$, the order of* $\mathrm{Alt}(q)$ *does not divide the order of the affine linear group* $\mathrm{AGL}(d, p)$.

Proof. Let $r = 3$ if $p = 2$ and let $r = 2$ otherwise. The result follows from Lemma 4.3 (except for the two easy cases $p = 2, d \leq 3$). □

Proof of Theorem 4.2*, part* (2). We show that the alternating groups constructed by the procedure are in the socle of G and G has a unique minimal normal subgroup. These facts imply that the G-action on D has a trivial kernel. We say that the group H is *involved* in the group K if $H \cong L/M$ for some $M \triangleleft K, M \leq L \leq K$. If a simple group H is involved in K then clearly H is involved in a composition factor of K.

We may assume G is not a giant. Let K be the kernel of the G-action on D. The stabilizer of E_1 restricted to E_1 passed TEST_GIANT, whence it contains $\mathrm{Alt}(q)$, $q := |E_1|$. As the G-action on the set of blocks is transitive, the same holds for each E_i. Also, it follows that $\mathrm{Alt}(q)$ is involved in G/K.

If $\mathrm{Soc}(G)$ is abelian then, by Theorem 2.6, $m = p^d$ for some prime p and $G \leq \mathrm{AGL}(d, p)$. But, $d \log p = \log\ m \leq \log\ n \leq q/4$ and therefore, by Corollary 4.4, the order of $\mathrm{Alt}(q)$ does not divide $|G|$. Thus this case cannot occur. Hence $\mathrm{Soc}(G)$

is nonabelian and the results stated in Theorem 2.7 apply. We use the notation of Theorem 2.7 and refer to $N \triangleleft G$ established there.

First we show that $\mathrm{Alt}(q)$ is not involved in $G/\mathrm{Soc}(G)$. Indeed, otherwise $\mathrm{Alt}(q)$ must be involved either in G/N or in $N/\mathrm{Soc}(G)$. The first case is impossible because $G/N \leq S_d$ (Theorem 2.7(b)) and $d \leq \log\ m/\log\ 5 < q/8$ (Theorem 2.7(c)). In the second case, $\mathrm{Alt}(q)$ is involved in $N/\mathrm{Soc}(G) \leq \mathrm{Out}(T)^d$, a solvable group by the Schreier conjecture (Theorem 2.3), again a contradiction.

It follows now that $\mathrm{Alt}(q)$ is involved in $\mathrm{Soc}(G)$ and $K \not\geq \mathrm{Soc}(G)$. Now $\mathrm{Soc}(G)$ must be the unique minimal normal subgroup for otherwise, by Theorem 2.8, we have a contradiction:

$$n^2 \geq m^2 = |\mathrm{Soc}(G)| \geq |\mathrm{Alt}(q)| = q!/2 > 2^q \geq n^4.$$

It follows that K contains no minimal normal subgroup, whence $K = 1$. □

Proof of Theorem 4.2*, part* (3). Assume the order of $|G|$ exceeds the stated bound. We must show that at no point will "small group" be falsely announced. This would not happen in Step 1, for $m \leq 3\log^2 n$ implies $|G| \leq m! < (3\log^2 n)^{3\log^2 n}$. If G is 2-transitive then the Step 2 call to TEST_GIANT will correctly halt with that revelation (by Theorem 2.4).

By Theorem 2.9, G is of Cameron type and A can be identified with the set of points of a Cameron scheme $C(m, k, s, r)$, and we may assume that $rs > 1$; that is, G is not a giant. Of course, the parameters and the identification are not known a priori. We prove that Step 3 of NATURAL_ACTION correctly recovers this structure with D corresponding to rB, E_i to B_i (so $q = k$), and the parameter k satisfies $k > 4\log n$. (We use the letters r, k, B_i, $C = rB = B_1 \cup \cdots \cup B_r$ to mean what they do in section 2.5. We call the action of G on C "natural." Recall that each $a \in A$ corresponds to an s-transversal $T(a) \subset rB$.)

We take note of some inequalities satisfied by the parameters of this $C(m, k, s, r)$. Since $m = \binom{k}{s}^r \geq (k/s)^{rs} \geq 2^{rs}$,

$$rs \leq \log m. \tag{4.1}$$

Since $rs > 1$, we have $m \geq \binom{k}{2}$ which implies

$$k < 2\sqrt{m}. \tag{4.2}$$

Also,

$$k > 4\log n; \tag{4.3}$$

otherwise, $|G| \leq (k!)^r r! \leq k^{kr} r! \leq m^k r! \leq n^{4\log n}(\log n)! < \exp(9\log^2 n\ \log\log\ n)$. Finally,

$$k \geq 2rs^2; \tag{4.4}$$

otherwise, using (4.1), we have $|G| \leq (k!)^r r! < (2rs^2)^{2r^2s^2} r! < (2\log^2 n)^{2\log^2 n}(\log\ n)!$ $< \exp(9\log^2 n\ \log\log\ n)$.

We claim now that the G-action on D is similar to the natural G-action on C. For $b \in rB$, let $U(b) = \{u \in A | b \in T(u)\}$. We need to show that that $D = \{U(b) | b \in rB\}$. By (4.4) and Lemma 2.10, $\Gamma = \Sigma_1$ and $\Delta = \Phi$. Thus, for any $y \in \Gamma(x)$, the set $T(x) - T(y)$ is a singleton $\{b(x, y)\}$. Now, a simple inspection of the Cameron scheme,

using that $k > 3s$ (by (4.4) since $rs > 1$), shows that $C(x, y) = U(b(x, y))$. The result follows since G acts transitively on C.

The identification of the E_i with the B_i follows because the latter are the unique minimal-size blocks in the natural action of G. Finally, (4.1), (4.2), and (4.3) ensure that the cardinality tests on that $\Gamma(x), D(x), D, E_i$ in Steps 3 and 4 do not cause terminal output "small group." □

4.2. Time complexity of the subroutine NATURAL_ACTION.

LEMMA 4.5. *Suppose $G = \langle Q \rangle$ acts on an m-set, and $|Q| = q$. Then TEST_GIANT(G) runs in $O^{\sim}(m^2 + qm)$ time.*

Proof. We work with $O^{\sim}(1)$ coset representative sets R_i each of size $2 \leq |R_i| \leq m$, and, by Theorem 2.12(b), $O^{\sim}(1)$ sets of generators T_i of size $O^{\sim}(1)$. Hence the sifting of products of the form $\rho\tau$, $\rho \in R_i, \tau \in T_{i-1}$ for some i costs $O^{\sim}(m^2)$; in addition, we may have to sift the elements of Q. □

THEOREM 4.6. *Suppose $G = \langle Q \rangle$ acts on an m-set, and $|Q| = q$. Then NATURAL_ACTION(Q) runs in $O^{\sim}(qm^2)$ time.*

Proof. Testing 2-transitivity takes $O(m^2q)$ time. Hence, by Lemma 4.5, Step 2 can be executed in $O^{\sim}(m^2q)$. Finding the orbitals requires $O(m^2q)$ steps. The (local) computation of $\{\alpha(w) : w \in A\}$ requires $O^{\sim}(m^2 + qm)$ time. The number of Schreier generators is qm; they are found in $O(m^2q)$ time. $O^{\sim}(qm^{3/2})$ steps suffice to find the $\beta(y')$. $C(x, y)$ can be determined in $O(m^2)$ time. We compute $D(x)$ in $O^{\sim}(m^2)$. Finally, D is also obtained in $O^{\sim}(m^2)$. Thus Step 3 requires $O^{\sim}(m^2q)$ total time.

The action of any $\phi \in Q$ on D can be found in $O^{\sim}(m^{3/2})$. The structure tree is computed in $O^{\sim}(mq)$, and, since $r = O^{\sim}(1)$, generators for the stabilizers of orbit-representatives on the new tree can be computed in $O^{\sim}(qm)$ time. Finally, we call TEST_GIANT on a set of size $O^{\sim}(\sqrt{m})$, with $O^{\sim}(q)$ generators, requiring $O^{\sim}(m + q\sqrt{m})$ time. Thus the time complexity of Step 4 is $O^{\sim}(mq + m^{3/2})$. □

COROLLARY 4.7. *Step 2 of the main algorithm runs in $O^{\sim}(sn^2)$ time.*

Proof. We apply NATURAL_ACTION to the action $G(v)$ of the point stabilizer G_v on the children of v for certain nodes v of the structure forest (one node from each level of each tree). Denoting by q_v the number of (Schreier) generators for G_v and by m_v the number of children of v, $\sum_v (q_v m_v) = O(sn)$. □

4.3. Extending the structure forest.

PROPOSITION 4.8. *Step 3 of the main algorithm runs in $O^{\sim}(sn^2)$ time.*

Proof. Suppose that a node v is the representative of an orbit in the original SF, v has m children, and NATURAL_ACTION appended a tree $T(v)$ to v. The vertices of $T(v)$ are subsets of the children of v; hence the group elements carrying v to the other nodes of its orbit v^G, computed in Step 1, naturally define a copy of $T(v)$ appended to the other nodes in v^G. These copies of $T(v)$ can be obtained in $O^{\sim}(m^{3/2}|v^G|)$ and the action of any $\sigma \in G$ can be extended to the appended trees within the same time bound. Hence the action of σ on the entire ESF can be computed in $O^{\sim}(n^{3/2})$. The extension to the SD is straightforward and in time $O(n)$. Finally, as in Lemma 3.1(b), generators of G_v for representatives of G-orbits of the SD can be constructed in $O(sn^2)$ time. □

5. Constructing strong generators for a giant. The purpose of this section is to construct a Jerrum-style compact SGS for the giants. Recall that the "giants" are the symmetric and alternating groups in their natural action. The Jerrum-style compact SGS for $G = \mathrm{Sym}(C)$ acting on the set $C = \{x_1, x_2, \ldots, x_m\}$ consists of $m-1$ permutations $\pi_1, \pi_2, \ldots, \pi_{m-1}$ such that for all $1 \leq k \leq m-1$, π_k fixes $x_1, \ldots, x_{k-1}$

and $x_k^{\pi_k} = x_{k+1}$. For $G = \mathrm{Alt}(C)$, the SGS contains $\pi_1, \pi_2, \ldots, \pi_{m-2}$ as above while π_{m-1} fixes $x_1, \ldots, x_{m-3}$ and $x_{m-2}^{\pi_{m-1}} = x_m$.

We require that the strong generators are constructed from the given generators of the giant by the following *legal operations: multiplication, inversion, and taking powers of permutations*. The reason for this constraint is that the procedure is applied to the case when $G(v)$, the action of the stabilizer of node v on the children in the structure domain, is an alternating group. In this case, although we know a priori that some $\sigma \in G(v)$ acts on the children as, e.g., a given 3-cycle, no such permutation can be guaranteed to belong to the input group unless it has been constructed, by way of legal operations, from the generators of $G(v)$. In this application, all group operations are global; i.e., we perform them on all points in the SD.

5.1. Construction of a 3-cycle. The essence of the procedure is the construction of a 3-cycle. Once a 3-cycle ρ is constructed, an SGS can be obtained easily by taking appropriate conjugates of ρ.

We note that a byproduct of the procedure yields a simple, elementary proof of the old result, known to Jordan (1895) [Jo] (and vastly surpassed by Theorem 2.4) that the only $c \log^2 n / \log \log n$ -fold transitive permutation groups are the giants [BS87]. It also yields an $\exp(\sqrt{n \ln n}(1 + o(1)))$ upper bound on the diameter of any Cayley graph of the giants [BS88].

Our goal is achieved by the procedure THREE_CYCLE. As a preprocessing phase, we determine and store the first $\log n$ primes. (The global variable n is the degree of the permutation group which is the input of the entire algorithm; we assume that n is sufficiently large.) We denote the ith prime by p_i and the product of the first i primes by $p(i)$.

Also, we need the following definitions. For $\pi \in \mathrm{Sym}(C)$, let us call a subset B of $\mathrm{supp}(\pi)$ *independent* with respect to π if $B \cap B^\pi = \emptyset$. The *commutator* of $\pi, \tau \in \mathrm{Sym}(C)$ is $[\pi, \tau] = \pi\tau\pi^{-1}\tau^{-1}$.

The procedure THREE_CYCLE uses the subroutine ORBITALS. Given generators for some $G \leq \mathrm{Sym}(C)$, $|C| = m$, ORBITALS returns $O(\log m)$ generators for a subgroup $H \leq G$ with the same orbitals as G. In particular, if G is a giant then ORBITALS returns $O(\log m)$ generators for a 2-transitive subgroup. The idea is the following. Suppose that generators for some $H \leq G$ are already defined but the orbital structures of the two groups are different. We fix an ordering of the generators $P = \{\tau_1, \ldots, \tau_k\}$ of G, and, for each H-orbital Δ which is not an orbital of G, we find the last element of P which moves Δ. Then we add a product of the form $\tau_1^{\varepsilon_1}\tau_2^{\varepsilon_2}\cdots\tau_k^{\varepsilon_k}$ to the generators of H where each $\varepsilon_i \in \{0, 1\}$ and ε_j is chosen such that $\tau_1^{\varepsilon_1}\tau_2^{\varepsilon_2}\cdots\tau_j^{\varepsilon_j}$ moves at least half of the H-orbitals Δ for which τ_j was the last generator moving Δ. This is a deterministic version of the *random subproduct method*, which we describe in section 5.4.

procedure ORBITALS(P, C, R)
INPUT:$G = \langle P \rangle \leq \mathrm{Sym}(C)$, $|C| = m$, $P = \{\tau_1, \ldots, \tau_k\}$.
OUTPUT: Generators R for some $H \leq G$ with same orbitals as G, $|R| = O(\log m)$.
Initialize: compute orbitals $O_1, \ldots, O_p$ of G; $R = \emptyset$
repeat
    compute orbitals $\{\Delta_i : i \in I\}$ of $\langle R \rangle$
    for all $\Delta_i \notin \{O_1, \ldots, O_p\}$, compute
        $\mathrm{last}(\Delta_i) := \max\{j : \Delta_i^{\tau_j} \neq \Delta_i\}$
    $\sigma := 1$ ($*$ start constructing new element of R $*$)

for $j := 1$ to k **do**
if $|\{\Delta_i : \text{last}(\Delta_i) = j, \Delta_i^\sigma = \Delta_i\}| > |\{\Delta_i : \text{last}(\Delta_i) = j, \Delta_i^\sigma \neq \Delta_i\}|$
then $\sigma := \sigma\tau_j$
$R := R \cup \{\sigma\}$
until orbitals of G = orbitals of $\langle R\rangle$
end (ORBITALS).

Steps 1–3 in the next procedure can be viewed as a preprocessing phase in which we construct the first $\log^2 n$ coset representative sets in the stabilizer chain of a giant G. With these coset representative sets in hand, it is easy matter to construct a permutation $\tau \in G$ that has a prescribed effect on an arbitrary subset of size $\log^2 n$ (cf. Lemma 5.2). Such constructed elements are useful in a computation that replaces a given element λ by one with significantly smaller support. For an appropriately designed τ, $\lambda_1 = [\lambda, \tau]$ contains cycles of prime length for a lot of different primes. An underlying idea then is that one of these primes does not divide most of the cycle lengths in λ_1. Taking an appropriate power of λ_1, we can kill all cycles whose length was not divisible by that prime and we get a permutation with smaller support. Iterating the process, we obtain a 3-cycle.

procedure THREE_CYCLE(Q)
INPUT: $G = \langle Q\rangle$ acting on $C = \{x_1, x_2, \ldots, x_m\}$; $m > 3\log^2 n$, G^C is a giant.
OUTPUT: An SGS, constructed from Q (using legal operations only).
Step 1. Begin PERMREP(Q, C);
stop PERMREP(Q, C) when $|\{i : i > \log^2 n, |R_i| \neq 1\}| = 2\log n + \log^2 n$.
Let R be a collection of nontrivial coset representatives such that
$|R \cap R_i| = 1$ for all $i > \log^2 n$, $|R_i| \neq 1$.
Step 2. ORBITALS(Q, C, Q_0).
Step 3. **for** $i := 1$ to $\log^2 n$ **do**
Let $G(i-1) := \langle Q_{i-1} \cup R\rangle$.
Construct coset representatives D_i for $G(i-1)^C$ mod $G(i-1)^C_{x_i}$
($*$ $G(i-1)^C_{x_i}$ is the stabilizer of x_i in $G(i-1)^C$. $*$)
Construct Schreier generators Q_i^* for $G(i-1)^C_{x_i}$.
ORBITALS(Q_i^*, C, Q_i).
Step 4. Compute $f(m), g(m)$, where $f(m) := \min\{\{\}r : p(r) > m^4\}$, and $g(m) := \sum_{i=1}^{f(m)} p_i$.
Step 5. Let λ be any $\neq 1$ element of G.
while $\deg(\lambda) > \log^2 n$ **do**
Choose $B \subset \text{supp}(\lambda), |B| = g(m)$ such that B is independent.
Construct $\tau \in G$ such that τ fixes pointwise B^λ and
$\tau|_B$ consists of cycles of length $p_1, p_2, \ldots, p_{f(m)}$.
$\lambda_1 := [\lambda, \tau]$.
For all $i \leq f(m)$, compute $m(i) :=$ the product of all cycle lengths in λ_1
which are not divisible by p_i.
Choose $i \leq f(m)$ such that $2 \leq \deg(\lambda_1^{m(i)}) < \deg(\lambda)/2$.
Let $\lambda := \lambda_1^{m(i)}$ for this i.
end
Step 6. Construct $\rho \in G$ such that ρ fixes exactly $\deg(\lambda) - 1$ points in $\text{supp}(\lambda)$.
Let $\sigma = [\lambda, \rho]$. ($*$ σ is a 3-cycle $*$)
Step 7. Take conjugates of σ to obtain an SGS for $\text{Alt}(C)$.
If $G^C = \text{Sym}(C)$ then sift an odd permutation to obtain an SGS for G^C.

output the SGS for G^C.
end (THREE_CYCLE).

5.2. Correctness of the subroutine THREE_CYCLE.

LEMMA 5.1. *For each* $1 \le i \le \log^2 n$, $\langle Q_i^* \rangle^{C \setminus \{x_1,\ldots,x_i\}}$ *contains* $\mathrm{Alt}(C \setminus \{x_1,\ldots,x_i\})$.

Proof. It is proved by Theorem 2.4. □

LEMMA 5.2. *Given any* $D \subset C, |D| = d \le \log^2 n$, *and an injection* $f : D \to C$, *it is possible to construct* $\tau \in G$ *such that* $\tau|_D = f$, *and* τ *is a* $2d$*-long product of elements of* $\bigcup_{i=1}^{\log^2 n} D_i$.

Proof. By Lemma 5.1, for all $i \le \log^2 n$ $D_i = \{\alpha(i,j) | i \le j \le m\}$, where $\alpha(i,j)$ fixes $x_1, x_2, \ldots, x_{i-1}$ and moves x_i to x_j. For any distinct $a_1, \ldots, a_d \in C$, let us define recursively $\pi(a_1,\ldots,a_d) = \rho\alpha(d, a_d^\rho)^{-1}$, where $\rho = \pi(a_1,\ldots,a_{d-1})$. Then, for $i \le d$ we have $a_i^{\pi(a_1,\ldots,a_d)} = i$. Let now $D = \{l_1,\ldots,l_d\}$. Then $\tau = \pi(l_1,\ldots,l_d)\pi(f(l_1),\ldots,f(l_d))^{-1}$ is appropriate. □

PROPOSITION 5.3. $f(m) = O(\frac{\log\ m}{\log\ \log\ m})$ *and* $g(m) = O(\frac{\log^2 m}{\log\ \log\ m})$.

Proof. It is proved by the prime number theorem [HW]. □

COROLLARY 5.4. $f(m) = O(\log n)$ *and* $g(m) = O(log^2 n)$.

The following is easily verified.

LEMMA 5.5. *Let* $\pi, \tau \in \mathrm{Sym}(C)$. *Assume that* B *is an independent set with respect to* π *and* $\tau|_{B^\pi}$ *is the identity. Then* $[\pi,\tau]|_B = \tau^{-1}|_B$.

LEMMA 5.6. *Let* $\pi \in \mathrm{Sym}(m)$, $k = \deg(\pi)$. *Suppose* π *contains cycles of each prime length* p_i, $i \le r = f(m)$. *Let* $m(i)$ *be the product of all cycle lengths occurring in* π *which are not divisible by* p_i. *Then* $2 \le \deg(\pi^{m(i)}) < k/4$ *for some* $i \le r$.

Proof. Let $K = \mathrm{supp}(\pi)$. For each $x \in K$, let us consider the set $P(x)$ of those primes p_i dividing the length of the π-cycle through x. Clearly, the product of these primes is $\le k$.

Let $n(i)$ denote the number of points x such that $p_i \in P(x)$. Let us estimate the weighted average W of the $n(i)$ with weights $\log\ p_i$. Recall that the sum of the weights is $\sum \log\ p_i > \log(m^4) = 4 \log\ m$; therefore,

$$W < \sum_{x \in K} \sum_{p_i \in P(x)} \log\ p_i/(4 \log\ m)$$

$$\le (k \log\ k)/(4 \log\ m) \le k/4.$$

We thus infer that $n(i) < k/4$ for some $i \le r$. Clearly, $\pi^{m(i)}$ is not the identity and it fixes all but $n(i)$ points. □

THEOREM 5.7. *The output of THREE_CYCLE*(Q) *is an SGS for* G.

Proof. By Lemma 5.1, Step 3 constructs the first $\log^2 n$ coset-representative sets for a giant. By Corollary 5.4, we can choose independent sets of size $g(m)$ from permutations of degree $> \log^2 n$. By Lemma 5.2, we are able to construct the permutations τ, ρ required in Step 5 and 6. By Lemma 5.5, $\lambda_1|_B$ has the same cycle structure as $\tau|_B$; moreover, $\deg(\lambda_1) \le \deg(\lambda) + \deg(\tau\lambda^{-1}\tau^{-1}) = 2\deg(\lambda)$. Hence, by Lemma 5.6, we can choose $i \le r$ such that $2 \le \deg(\lambda_1^{m(i)}) < \deg(\lambda)/2$. In Step 6, we compute the commutator of two permutations whose supports intersect in exactly one point, whence the commutator is a 3-cycle. Finally, we can obtain permutations which conjugate σ into elements of an SGS by Lemma 5.2. □

5.3. Time complexity of THREE_CYCLE.

LEMMA 5.8. *ORBITALS(P, C, R) runs in $O^\sim(|P||C|^2 + |P|n)$ time.*

Proof. The orbitals of G can be computed in $O(|P||C|^2)$ time. One execution of the repeat loop costs $O^\sim(|R||C|^2)$ for the computation of the orbitals of $\langle R\rangle$, plus $O(|P||C|^2)$ for the computation of the function $\text{last}(\Delta_i)$, plus $O(|C|^2)$ for checking the images Δ_i^σ, plus $O(|P|n)$ for group multiplications to compute σ. The key observation is that we execute the repeat loop only $O(\log|C|)$ times since, at each execution, the new σ increases at least half of the orbitals Δ_i for which the function $\text{last}(\Delta_i)$ is defined. Therefore, after l executions of the repeat loop, the number of "bad" Δ_i's is $\le |C|^2(3/4)^l$. □

LEMMA 5.9. *Suppose that the sum of the different positive integers b_i is $\le m$. Then $\prod b_i \le \exp(O^\sim(\sqrt{m}))$.*

Proof. Choose $b_1 < b_2 < \cdots$ such that $\prod b_i$ is maximal. Then $b_1 \le 4$, for otherwise substituting b_1 by 2 and $b_1 - 2$ the product would increase. Also, for any i, $b_{i+1} - b_i \le 2$; otherwise the product would increase by substituting $b_i + 1$ and $b_{i+1} - 1$ for b_i and b_{i+1}. Moreover, $b_{i+1} = b_i + 2$ for at most one i: if $b_{i+1} \ge b_i + 2$ and $b_{j+1} \ge b_j + 2$ for some $i < j$ then by substituting b_i by $b_i + 1$ and b_{j+1} by $b_{j+1} - 1$ the product would increase. Thus the b_i comprise an initial segment of the natural numbers with the possible omission of 1, 2, 3 and one other number. If $\max\{b_i\} = x$, then $m \ge \sum b_i \ge x(x+1) - 1 - 2 - 3 - (x-1)$, which implies $x \le 2 + \sqrt{2m}$. We have then $\prod b_i < x! = \exp(O^\sim(\sqrt{m}))$. □

THEOREM 5.10. *Suppose that $|Q| = q$ and $\langle Q\rangle^C$ is a giant. Then THREE_CYCLE(Q) constructs an SGS for $\langle Q\rangle^C$ in $O^\sim(qn + mn + m^2q + m^3)$ time.*

Proof. By the prime number theorem, the $\log n$th prime is $O(\log n \log\log n)$; hence the preprocessing phase requires $O^\sim(1)$ time. By Theorem 2.12(b) and the argument already used at the analysis of TEST_GIANT (cf. Lemma 4.5), Step 1 runs in $O^\sim(qn + mn)$. By Lemma 5.8, Step 2 requires $O^\sim(m^2q + qn)$ time and the output Q_0 satisfies $|Q_0| = O(\log m)$. We execute the loop of Step 3 $O^\sim(1)$ times. The coset representative set D_i is obtained in $O(mn)$. The Schreier generators are constructed in $O^\sim(mn)$ time and their number is $|Q_i^*| \le m|Q_{i-1}| = O^\sim(m)$. Using Lemma 5.8 again, Q_i is computed in $O^\sim(m^3 + mn)$; hence the total time requirement of Step 3 is $O^\sim(mn + m^3)$. Step 4 runs in $O^\sim(1)$. Since we decrease the degree of λ at least by a factor 2, the loop of Step 5 is executed $O^\sim(1)$ times. By Lemma 5.2, τ, whence λ_1, is obtained in $O^\sim(n)$. By Lemma 5.9, $m(i)$ is a $\le O^\sim(\sqrt{m})$-digit number; thus, for all $i \le r$, $m(i)$ can be computed in $O^\sim(m)$ time [SS], and $\lambda_1^{m(i)}$ can be constructed in $O^\sim(n\sqrt{m})$. (For all x in the permutation domain, we have to divide $m(i)$ by the length of the cycle through x.) Hence Step 5 requires $O^\sim(n\sqrt{m})$ time. Step 6 runs in $O^\sim(n)$. Finally, by Lemma 5.2, Step 7 requires $O^\sim(mn)$ time. □

COROLLARY 5.11. *Step 4 of the main algorithm runs in $O^\sim(n^3 + sn^2)$ total time.*

Proof. We apply THREE_CYCLE to the action of the stabilizer of some nodes v on the children of v in the SD. As in the proof of Corollary 4.7, denoting by q_v the number of (Schreier) generators for G_v and by m_v the number of children of v, $\sum_v (q_v m_v) = O(sn)$. □

5.4. Las Vegas speedup of THREE_CYCLE. In this section we present a randomized version of THREE_CYCLE with $O^\sim((q+m)n)$ running time. As indicated in the proof of Theorem 5.10, calls to the subroutine ORBITALS were the only parts of the procedure THREE_CYCLE not executable within this tighter time bound. ORBITALS is accelerated by using random subproducts of generators.

DEFINITION 5.12. *Let $G = \langle \tau_1, \tau_2, \ldots, \tau_k \rangle$. A random subproduct of the generators $\tau_1, \ldots, \tau_k$ is an instance of the product $\tau_1^{\varepsilon_1}\tau_2^{\varepsilon_2}\cdots\tau_k^{\varepsilon_k}$ where the ε_i are independent,* 0-1 *valued random variables with* $\mathrm{Prob}(\varepsilon_i = 0) = \mathrm{Prob}(\varepsilon_i = 1) = 1/2$.

The key observation is that a random subproduct of the generators is just as likely to increase an orbital of a subgroup $H \leq G$ as the deterministically constructed element σ in ORBITALS. We make this observation more precise in the following lemma.

LEMMA 5.13. *Let $G = \langle \tau_1, \tau_2, \ldots, \tau_k \rangle \leq \mathrm{Sym}(m)$. Then the expected number of random subproducts of the generators $\tau_1, \ldots, \tau_k$ which generate a subgroup H with the same orbitals as G is $c \log m$.*

Proof. Let H_t be the subgroup generated by the first t random subproducts and let $\sigma = \tau_1^{\varepsilon_1}\tau_2^{\varepsilon_2}\cdots\tau_k^{\varepsilon_k}$ be the $(t+1)$st random subproduct. Let $\{\Delta_i : i \in I\}$ be the orbitals of H_t which are not orbitals in G. For an arbitrary Δ_i, let $l = \mathrm{last}(\Delta_i) = \max\{j : \Delta_i^{\tau_j} \neq \Delta_i\}$. Then

$$\begin{aligned}
\mathrm{Prob}(\Delta_i^{\sigma} \neq \Delta_i) &= \mathrm{Prob}(\Delta_i^{\tau_1^{\varepsilon_1}\cdots\tau_l^{\varepsilon_l}} \neq \Delta_i) \\
&\geq \mathrm{Prob}(\varepsilon_l = 1 | \Delta_i^{\tau_1^{\varepsilon_1}\cdots\tau_l^{\varepsilon_{l-1}}} = \Delta_i)\mathrm{Prob}(\Delta_i^{\tau_1^{\varepsilon_1}\cdots\tau_l^{\varepsilon_{l-1}}} = \Delta_i) \\
&\quad + \mathrm{Prob}(\varepsilon_l = 0 | \Delta_i^{\tau_1^{\varepsilon_1}\cdots\tau_l^{\varepsilon_{l-1}}} \neq \Delta_i)\mathrm{Prob}(\Delta_i^{\tau_1^{\varepsilon_1}\cdots\tau_l^{\varepsilon_{l-1}}} \neq \Delta_i) \\
&= 1/2.
\end{aligned}$$

Hence, with probability $\geq 1/2$, σ enlarges each "bad" orbital of H_t. A standard argument shows that after taking t random subproducts the expected number of "bad" orbitals is $\leq m^2(3/4)^t$. □

The speedup of THREE_CYCLE is straightforward: instead of calling ORBITALS, we take $O(\log m)$ random subproducts of generators. The procedure is Las Vegas since we can check in $O^{\sim}(m^2)$ time whether these random subproducts generate a 2-transitive group.

Chronologically, the idea of random subproducts preceded the subroutine ORBITALS (cf. [BLS88]). Random subproducts are useful far beyond the scope of this paper; for example, in [BCFLS91], [BCFLS95], augmented with other ideas, they provide a $O^{\sim}(n^3)$ *elementary* Monte Carlo SGS construction.

6. Descending the structure domain: Traversing levels. In the previous sections, we discussed the first four (preparatory) steps of the main algorithm. We constructed an extension of the original permutation domain, called the SD, and an ordered partition of the SD such that the pointwise stabilizer G_i of the first i sets is normal in G. The algorithm proceeds by constructing an SGS for successive G_i mod G_{i+1} and finding *normal generators* for G_{i+1} (that is, generators of subgroup whose normal closure in G is G_{i+1}). We describe the construction of these elements through a process of *normal sifting*, which relies on knowledge of presentations for the quotients G_i/G_{i+1}. Our time bounds depend critically on the number of normal generators obtained and, to that end, we indicate how we form concise presentations.

Recall that a *presentation* of a group G is a pair $\langle X \mid \mathcal{R} \rangle$, in which X is a set and $\mathcal{R} \subseteq \mathcal{F}(X)$ ($\mathcal{F}(X)$ denotes the free group on X) such that there is an epimorphism $\phi : \mathcal{F}(X) \to G$ with kernel $\langle \mathcal{R}^{\mathcal{F}(X)} \rangle$. We shall say that the presentation is *induced by* ϕ; in the algorithmic application of presentations, it is typically necessary to specify ϕ along with X and $\mathcal{R}$. The elements of $\mathcal{R}$ are called *relators*.

6.1. Normal sifting. Let

$$G = G_0 \geq G_1 \geq \cdots \geq G_m = N \tag{6.1}$$

be a chain of normal subgroups of G. Let $S_i \subset G_i$ generate G_i mod G_{i+1}, i.e., $G_i = \langle S_i \rangle G_{i+1}$ $(i \leq m-1)$. We call the collection $\{S_i : 0 \leq i \leq m-1\}$ a system of *chain generators* of the series (6.1).

Suppose that

$$G_i/G_{i+1} = \langle X_i \mid \mathcal{R}_i \rangle \tag{6.2}$$

is a presentation of G_i/G_{i+1} induced by $\phi : X_i \to G_i/G_{i+1}$. We say that $S_i \subseteq G_i$ *corresponds* to this presentation if the natural map $G_i \to G_i/G_{i+1}$ yields a bijection $S_i \to \phi(X_i)$. Then, for $w(X_i) \in \mathcal{R}_i$, substitution of S_i for X_i yields an element $w(S_i) \in G_{i+1}$.

Assume that the subgroup chain

$$G = H_0 \geq H_1 \geq \cdots \geq H_f = N \tag{6.3}$$

is a refinement of (6.1): $G_i = H_{j_i}$ $(0 = j_0 < j_1 < \cdots < j_m = f)$. Assume further that a set C_j of right coset representatives of H_{j-1} mod H_j is given for each j, $1 \leq j \leq f$ such that for $j_i + 1 \leq j \leq j_{i+1}$, we have $C_j \subset \langle S_i \rangle$. Such a system will be called *compatible* with the given system $\{S_i\}$ of chain generators of (6.1). Given an element of $g \in G$, we can sift it down along the chain $\{H_j\}$ to obtain a siftee, a member of N. This defines the map sift: $G \to N$.

THEOREM 6.1 (normal sift theorem). *Assume a series of normal subgroups* (6.1) *of the group $G = \langle S \rangle$ is given along with chain generators $\{S_i \mid 0 \leq i \leq m-1\}$ which correspond to presentations* (6.2) *of the factors. Assume a refinement* (6.3) *of* (6.1) *is given along with coset representatives, compatible with the given chain generators. Let Q denote the set of the following elements:*

(a) *S (the set of generators of G);*

(b) *$g^{-1}hg$ for $g \in S$ and $h \in S_i$, $1 \leq i \leq m-1$;*

(c) *$w_i(S_i)$ for all $w_i \in \mathcal{R}_i$, $0 \leq i \leq m-1$.*

Then $N = \langle \mathrm{sift}(Q)^G \rangle$.

Proof. Let $H = \langle \mathrm{sift}(Q)^G \rangle$. Set $\bar{G} = G/H$ and let $\phi : G \to \bar{G}$ be the natural homomorphism. Clearly, $H \leq N$, and therefore $|\bar{G}| \geq |G/N|$. We have to prove that equality holds here. For any subset $U \subset G$, we use $\bar{U}$ to denote $\phi(U)$.

Let $H_i = \langle S_i, S_{i+1}, \ldots, S_{m-1} \rangle$. $(H_m = 1.)$

1. $\bar{H}_0 = \bar{G}$, because $\mathrm{sift}(S) \subset H$ by (a).

2. $\bar{H}_i \triangleleft \bar{G}$, because $\mathrm{sift}(S_i^S) \subset H$ by (b).

3. $|\bar{H}_i/\bar{H}_{i+1}| \leq |G_i/G_{i+1}|$, because $w_i(\bar{S}_i) \in \bar{H}_{i+1}$ for $w_i \in \mathcal{R}_i$ by (c).

It follows that $|\bar{G}| = |\bar{H}_0/\bar{H}_1| \cdots |\bar{H}_{m-1}/\bar{H}_m| \leq |G_0/G_1| \cdots |G_{m-1}/G_m| = |G/N|$. □

6.2. Presentations. The normal sift theorem is applied each time our descent of the structure domain finishes a level. There, we are dealing with quotients G_i/G_{i+1} that act faithfully on L_{i+1}, the $(i+1)$st level of the SD. For our time bounds, we need to ensure that $|\mathcal{R}_i| = O^{\sim}(|L_{i+1}|^2)$.

For a full alternating group, $\mathrm{Alt}(q)$, there is a concise set of at most q relations [Car], cf. [CM, p. 67]. We quote Carmichael's presentation of $\mathrm{Alt}(q)$.

THEOREM 6.2 (see [Car]). *Fix $q \geq 4$. Let $X = \{x, y\}$. Let*

$$\mathcal{R}_{\mathrm{Car}} = \{y^{q-2}, x^3, (yx)^q\} \cup \{(xy^{-k}xy^k)^2 \mid 1 \leq k \leq (q-3)/2\}$$

if q is odd, and

$$\mathcal{R}_{\mathrm{Car}} = \{y^{q-2}, x^3, (yx)^{q-1}\} \cup \{(x^{(-1)^k} y^{-k} x y^k)^2 \mid 1 \le k \le (q-2)/2\}$$

if q is even. Then $\langle X \mid \mathcal{R}_{\mathrm{Car}} \rangle$ is a presentation of $\mathrm{Alt}(q)$.

This extends easily to a presentation of direct products of alternating groups, the situation we uncover at alternating levels of the SD. We use a Carmichael presentation, with a pair of generators, for each factor and enter the relators (commutators) that ensure that the pairs of generators commute.

For the small-group levels, we recall an elementary construction of presentations. Suppose that, for $1 \le j \le f$, C_j is a complete set of right coset representatives for $H_{j-1} \bmod H_j$, where

$$G = H_0 \ge H_1 \ge \cdots \ge H_f = 1.$$

For each $\gamma \in \bigcup_{j=1}^{f} C_j$, associate a symbol x_γ and let X be the collection of these symbols. For any $j \ge k$ and $1 \ne \sigma \in C_j$, $1 \ne \tau \in C_k$,

$$\sigma\tau = \gamma_f \cdots \gamma_{j+1}\gamma_j, \text{ for unique } \gamma_p \in C_p,\ j \le p \le f.$$

Let $w_{\sigma,\tau}$ be the word $x_\tau^{-1} x_\sigma^{-1} x_{\gamma_f} \cdots x_{\gamma_{j+1}} x_{\gamma_j}$ and let $\mathcal{R}$ be the collection of all such words. Then $\langle X \mid \mathcal{R} \rangle$ is a presentation of H.

Let $H = G_i/G_{i+1}$ be a small-level group acting on L_{i+1}, $|L_{i+1}| = m$. Coset representatives in the point stabilizer chain for H are available via PERMREP (Proposition 2.11). We know, however, that H is contained in a direct product of isomorphic primitive groups, this direct product acting as a "small" group on each of its, say r, orbits each of size m/r. Any such orbit includes at most $O^\sim(1)$ points where the point stabilizer chain for H decreases, i.e., where $|C_i| \ne 1$. Furthermore $|C_i| \le m/r$ for all i. It follows that $|X| = O^\sim(m)$ and $|\mathcal{R}| = O^\sim(m^2)$.

7. Descending the structure domain: Small group levels. By the results of section 4, the group $G(w)$ (the action of the stabilizer of the node w in the structure domain on the children of w), is either an alternating or a small group. (A small group is of order $< \exp(9 \log^2 n \log\log n)$.) Moreover, for $w, w' \in L_i$ these groups are isomorphic. We call L_i an *alternating level* if $G(w)$ is alternating for $w \in L_i$, and a *small group level* in the other case. Our objective in this section is to get past a small group level L_{i-1}. Suppose that we have constructed an SGS for G/G_{i-1} and normal generators Q^{i-1} for G_{i-1}. We proceed to constructing an SGS for G_{i-1}/G_i and normal generators Q^i for G_i.

The routine NORMCL(Q^{i-1}, L_i, S) gives us the SGS. A presentation for G_{i-1}/G_i is obtained according to section 6.2, and then normal generators for G_i are constructed according to Theorem 6.1.

Timing analysis for NORMCL. Let $L_{i-1} := \{w_1, w_2, \ldots, w_r\}$, and denote by B_j the children of w_j. Then $L_i = \bigcup_{1 \le j \le r} B_j$ and $|(G_{i-1})^{L_i}| = \exp(O^\sim(|L_{i-1}|))$. Moreover, since G_{i-1} stabilizes L_{i-1} pointwise, $t := \max(|(G_{i-1})_j^{L_i} : (G_{i-1})_{j+1}^{L_i}|) \le |L_i|/r$. (Recall that $(G_{i-1})_j^{L_i}$ denotes the jth subgroup in the pointwise stabilizer chain in the group G_{i-1} acting on the set L_i.) Therefore, by Theorem 2.15, the running time of NORMCL(Q^{i-1}, L_i, S) is $O^\sim\big(|L_{i-1}|n(|Q^{i-1}| + s|L_{i-1}| + |L_i||L_{i-1}|)\big)$.

Number of normal generators obtained. There are $O^\sim(|L_i|)$ coset representatives, so $|Q^i| \le |Q^{i-1}| + O^\sim(s|L_i| + |L_i|^2)$.

Finally we observe that the time to sift each normal generator into G_i is $O^\sim(n|L_{i-1}|)$.

Remark 7.1. If $s > n$ then we may apply NORMCL(Q^{i-1}, L_i, S^*) with $S^* := \bigcup_{j<i-1} S^j$. Since S^* is a set of compatible generators for G/G_{i-1} and Q^{i-1} contains the siftees of S into G_{i-1}, $\langle S\rangle = \langle S^*, Q^{i-1}\rangle$ and $\langle Q^{i-1}\rangle^{\langle S^*\rangle} = \langle Q^{i-1}\rangle^G$. This change improves the timing and the bound on the number of generators, replacing s by n in both expressions.

8. Descending the structure domain: Alternating levels. Suppose that we have constructed an SGS for G/G_{i-1} in Step 5 of the main algorithm and L_{i-1} is an alternating level. In this section, we describe a method to obtain an SGS for G_{i-1}/G_i and normal generators for G_i.

First, we introduce some notation. Let v be a representative node at level L_{i-1}. Level L_i can be partitioned into $L_i = B_1 \dot\cup B_2 \dot\cup \cdots \dot\cup B_r$, $|B_j| = m > 3\log^2 n$ for all j such that for each $w \in L_{i-1}$ the children of w comprise one of the B_j and the point stabilizer G_w acts as $\mathrm{Alt}(B_j)$ on this B_j. We may suppose that B_1 contains the children of v. While computing Schreier generators for G_v, the algorithm constructed $\alpha_2, \ldots, \alpha_r \in G$ such that $B_1^{\alpha_j} = B_j$. We denote by Q^{i-1} the set of normal generators for G_{i-1} constructed by the algorithm and by S the generators of G. Finally, for $\pi \in G_{i-1}$, $\mathrm{length}(\pi) := |\{j : \pi|_{B_j} \neq 1\}|$.

If all elements of Q^{i-1} act trivially on L_i, then $G_i = G_{i-1}$ and there is nothing to do. If there exists $\rho \in Q^{i-1}$ acting nontrivially on $B_{j'}$ for some $j' \leq r$, then, since G_{i-1} contains all conjugates of ρ, G_{i-1} acts as $\mathrm{Alt}(B_{j'})$ on $B_{j'}$; moreover, conjugating by $\alpha_2, \ldots, \alpha_r$ we see that $G_{i-1}^{B_j} = \mathrm{Alt}(B_j)$ for all $j \leq r$. Hence, by Proposition 2.2, G_{i-1}/G_i is isomorphic to $\mathrm{Alt}(m)^k$ for some k. We give an efficient version of Luks's "noncommutative linear algebra" to determine which coordinates of $\prod_{j\leq r} \mathrm{Alt}(B_j)$ are *linked* in the diagonal subgroups. We note that because of the transitive G-action on $\{B_1, \ldots, B_r\}$, the number of $\mathrm{Alt}(B_j)$'s is the same in each linked collection.

8.1. The procedure GIANT_CLOSURE. In Step 4 of the main algorithm, we computed an SGS $P_v \subseteq G$ for $G(v)$, the action of G_v on B_1. However, the elements of P_v are not necessarily in G_{i-1} (that is, they do not necessarily fix *all* nodes at level $i-1$). Here we describe a subroutine which computes an SGS $R \subseteq G_{i-1}$ for $\mathrm{Alt}(B_1)$ given P_v and given an element of G_{i-1} acting nontrivially on B_1.

More precisely, with additional applications in mind, we consider the following situation. The setwise stabilizer $G_{\{C\}}$ of a group G acts on a set C, $|C| \geq 8$, as $\mathrm{Alt}(C)$. The input to GIANT_CLOSURE is $P \subseteq G_{\{C\}}$ such that P is an SGS in this action, and $\rho \in G_{\{C\}}$. The output is R, an SGS for $\mathrm{Alt}(C)$, such that $R \subseteq \langle \rho^{\langle P\rangle}\rangle$; i.e., R is generated by conjugates of ρ by the elements of $\langle P\rangle$. Moreover, we require that there are $\tau, \sigma \in \langle \rho^{\langle P\rangle}\rangle$ such that $w(\tau, \sigma) = 1$ for $w(x, y) \in \mathcal{R}_{\mathrm{Car}}$ (see Theorem 6.2), and $R \subseteq \langle \tau, \sigma\rangle$.

If $C = \{1, 2, \ldots, m\}$, m even then $\tau = (1\ 2\ 3)$, $\sigma = (1\ 2)(3\ 4\ \ldots\ m-1\ m)$ satisfy the relations in Theorem 6.2. If m is odd then we can choose $\tau = (1\ 2\ 3)$, $\sigma = (3\ 4\ \ldots\ m-1\ m)$.

procedure GIANT_CLOSURE$(G, C, \rho, P, R, \tau, \sigma)$
INPUT:C, P, ρ as specified above.
OUTPUT: R, τ, σ.
Step 1. Let $\gamma_1 \in \langle P\rangle$ such that $\gamma_1|_C$ is a 3-cycle not commuting with $\rho|_C$. Compute $\rho_1 = [\rho, \gamma_1]$. $(*\ \deg(\rho_1|_C) \leq 6\ *)$ Take $\gamma_2 \in \langle P\rangle$ such that $|\mathrm{supp}(\rho_1|_C) \cap \mathrm{supp}(\gamma_2|_C)| = 1$. Compute $\rho_2 = [\rho_1, \gamma_2]$. $(*\ \rho_2|_C$ is a 3-cycle $*)$
Step 2. Conjugating ρ_2 with appropriate elements of $\langle P\rangle$, obtain permutations

$\pi_1, \pi_2, \ldots, \pi_{m-2}$ such that $\pi_i|_C = (i\ i+1\ i+2)$. $(*\ \pi_1, \pi_2, \ldots, \pi_{m-2}, \pi_{m-2}^2$ is an SGS for $\mathrm{Alt}(C)\ *)$.
Step 3. Compute τ, σ as specified before the procedure as a product of the π_i's.
Step 4. If m is odd then compute $\sigma\tau$. $(*\ \sigma\tau|_C = (1\ 2\ \ldots\ m-1\ m)\ *)$ R consists of τ and its conjugates by the powers of $\sigma\tau$.

If m is even then compute $\sigma\tau^2$ and $\tau\sigma^{-1}\tau\sigma$. $(*\ \sigma\tau^2|_C = (2\ 3\ \ldots\ m-1\ m)$ and $\tau\sigma^{-1}\tau\sigma|_C = (2\ 3\ 4)\ *)$ R consists of τ, $\tau\sigma^{-1}\tau\sigma$, and the conjugates of $\tau\sigma^{-1}\tau\sigma$ by the powers of $\sigma\tau^2$.
end (GIANT_CLOSURE).

PROPOSITION 8.1. *If group operations in G require $O(n)$ time then GIANT_CLOSURE$(G, C, \rho, P, R, \tau, \sigma)$ computes an SGS for* $\mathrm{Alt}(C)$ *in $O(mn)$.*

Proof. The correctness of the procedure is obvious. Given an SGS for $\mathrm{Alt}(C)$, any element of $\mathrm{Alt}(C)$ can be constructed from it by $O(m)$ group multiplications. Therefore, Steps 1 and 3 require $O(mn)$ time. By Lemma 5.2, a permutation with three prescribed positions can be constructed in $O(n)$ time so Step 2 also runs in $O(mn)$. Finally, we notice that using the result of the conjugation by the previous power of $\sigma\tau$ (or $\sigma\tau^2$), all conjugates in Step 4 can be computed with $O(m)$ group operations. □

8.2. The procedure GIANT_LINK. We obtain an SGS for G_{i-1}/G_i by applying the procedure GIANT_LINK. We use the notation introduced at the beginning of section 8 for the input; the output will be an SGS T and a set S^{i-1} of compatible generators for G_{i-1}/G_i and a set Q^i of normal generators for G_i.

If two coordinates $j, j' \le r$ are not linked in a diagonal action then there exists $\pi \in G_{i-1}$ such that $\pi|_{B_j} \neq 1$ and $\pi|_{B_{j'}} = 1$. In this case, we say that π *witnesses the separation of j from j'*. Note that possession of a witness to the separation of j from j' does not imply possession of a witness to the reverse separation, even though we know that one exists.

GIANT_LINK uses the subroutine GIANT_SEPARATE. The input is an SGS $R_j \subset G_{i-1}$ for $G_{i-1}^{B_j} \cong \mathrm{Alt}(B_j)$ and $\pi_1, \pi_2 \in G_{i-1}$ such that $\pi_l|_{B_j} \neq 1$ for $l = 1, 2$. The output is a single $\pi \in G_{i-1}$ such that, for any coordinate j', if either π_1 or π_2 witnesses the separation of j from j', then π also witnesses this separation.

procedure GIANT_SEPARATE(R_j, π_1, π_2, π)
INPUT:R_j, π_1, π_2 as specified above.
OUTPUT: π.
  if $\pi_1|_{B_j}, \pi_2|_{B_j}$ do not commute
    then $\pi := [\pi_1, \pi_2]$
    else take $\rho \in \langle R_j \rangle$ such that $\pi_1|_{B_j}, \rho^{-1}\pi_2\rho|_{B_j}$ do not commute
      $\pi := [\pi_1, \rho^{-1}\pi_2\rho]$
end (GIANT_SEPARATE).

PROPOSITION 8.2. *GIANT_SEPARATE computes the witness π in $O(n)$ time.*

Proof. The only nontrivial point is that an appropriate $\rho \in \langle R_j \rangle$ can be constructed in $O(n)$ time. If $\pi_1|_{B_j}$ has a fixed point, say $x^{\pi_1} = x$, then conjugate π_2 such that $x^{\rho^{-1}\pi_2\rho} = y$ for some y with $y^{\pi_1} \neq y$. If $\pi_1|_{B_j}$ does not have a fixed point then choose four different points $x, y, z, u \in B_j$ such that $x^{\pi_1} = y$ and $z^{\pi_1} = u$. Conjugate π_2 such that $y^{\rho^{-1}\pi_2\rho} = u$ and $x^{\rho^{-1}\pi_2\rho} \neq z$. Since we described the value of ρ at ≤ 4 points, such ρ can be obtained in $O(n)$ time by Lemma 5.2. Note that $\pi_2|_{B_{j'}} = 1$ implies $\rho^{-1}\pi_2\rho|_{B_{j'}} = 1$. □

In the first three steps of GIANT_LINK, we compute a subgroup of G_{i-1} which acts as the full alternating group on each B_j. In Step 4, we obtain witnesses for all pairs not linked by this subgroup and then compute a single witness for each B_j. In the loop described in Steps 5–7, we obtain additional elements of the subgroup G_{i-1} (until we have a collection that fully generates G_{i-1}/G_i). First, in Step 5, until the linked collections have the same length, we conjugate the shortest collection into all positions, necessarily breaking up some links in the longer collections. Step 7 ensures that we have done all the link breaking that is implied by the subgroup at hand and, if so, that the subgroup is normalized (mod G_i) by G; failure of either test produces a new witness to separation in Step 7, and the loop is repeated. If the tests are passed, we can specify Q^i (Step 8).

procedure GIANT_LINK$(L_i, Q^{i-1}, S, P_v, \{\alpha_2, \ldots, \alpha_r\}, Q^i, S^{i-1}, T)$
INPUT: $L_i = B_1 \dot\cup \cdots \dot\cup B_r, Q^{i-1}, S, P_v, \{\alpha_2, \ldots, \alpha_r\}$ as specified above.
OUTPUT: Q^i, S^{i-1}, T.
Step 1. take $\rho \in Q^{i-1}$, $\rho|_{B_j} \neq 1$ for some j; Compute $\rho_1 := \alpha_j \rho \alpha_j^{-1}$.
Step 2. GIANT_CLOSURE$(G, B_1, \rho_1, P_v, R_1, \tau_1, \sigma_1)$.
Step 3. **for** $j := 2$ to r **do**
compute $R_j := \alpha_j^{-1} R_1 \alpha_j, \sigma_j := \alpha_j^{-1}\sigma_1\alpha_j$, $\tau_j := \alpha_j^{-1}\tau_1\alpha_j$, and $\rho_j := \alpha_j^{-1}\rho_1\alpha_j$.
Step 4. **for** $j := 1$ to r **do**
Collect the following elements of G_{i-1} in a set Σ:
the siftees of $Q^{i-1} \cup \{\sigma_{j'}, \tau_{j'} : 1 \le j' \le r\}$ through R_j
$w(\tau_j, \sigma_j)$ for all $w(x, y) \in \mathcal{R}_{\mathrm{Car}}$ ($*$ see Theorem 6.2 $*$)
for $\sigma \in \Sigma$ **do**
for all coordinates j' for which σ witnesses the separation of j' from j but this separation is not witnessed by the current $\rho_{j'}$ **do**
GIANT_SEPARATE$(R_{j'}, \rho_{j'}, \sigma, \rho_{j'})$.
Step 5. **while** there exist j, j' with length$(\rho_j) \neq$ length$(\rho_{j'})$ **do**
take ρ_j with minimal length
for $j' := 1$ to r **do**
GIANT_SEPARATE$(R_{j'}, \rho_{j'}, \alpha_{j'}^{-1}\alpha_j\rho_j\alpha_j^{-1}\alpha_{j'}, \rho_{j'})$
if the lengths of all $\rho_{j'}$, $1 \le j' \le r$, are equal **and**
there exist j', j'' such that $\rho_{j''}$ witnesses a separation of j' (from some j''') that is *not* witnessed by $\rho_{j'}$
then for one such pair j', j''
GIANT_SEPARATE$(R_{j'}, \rho_{j'}, \rho_{j''}, \rho_{j'})$.
Step 6. **for** $j := 1$ to r **do**
GIANT_CLOSURE$(G, B_j, \rho_j, R_j, R_j, \tau_j, \sigma_j)$.
Step 7. **for** $j := 1$ to r **do**
Collect the following elements of G_{i-1} in a set Σ:
the siftees of $Q^{i-1} \cup \{\sigma_{j'}, \tau_{j'} : 1 \le j' \le r\}$ through R_j
$w(\tau_j, \sigma_j)$ for all $w(x, y) \in \mathcal{R}_{\mathrm{Car}}$
the siftees of $\{\sigma_{j'}^{\alpha}, \tau_{j'}^{\alpha} : \alpha \in S, 1 \le j' \le r\}$ through R_j
for $\sigma \in \Sigma$ **do**
for all coordinates j' for which σ witnesses the separation of j' from j but this separation is not witnessed by the current $\rho_{j'}$ **do**
GIANT_SEPARATE$(R_{j'}, \rho_{j'}, \sigma, \rho_{j'})$
if any of the ρ_j were changed in this step **then goto** Step 5.

Step 8. Let $J \subseteq \{1, 2, \ldots, r\}$ consist of a representative j from each linked collection of coordinates.
output $S^{i-1} := \{\tau_j, \sigma_j : j \in J\}$;
output $T := \bigcup\{R_j : j \in J\}$;
collect in Q^i the following elements of G_i:
the siftees of Q^{i-1} through the SGS T
for all distinct $j, j' \in J$, the commutators $[\sigma_j, \tau_{j'}], [\sigma_j, \sigma_{j'}], [\tau_j, \tau_{j'}]$, $[\tau_j, \sigma_{j'}]$
for all $j \in J$, $w(\tau_j, \sigma_j)$ for all $w(x, y) \in \mathcal{R}_{\mathrm{Car}}$
the siftees of $\{\alpha^{-1}\tau_j\alpha, \alpha^{-1}\sigma_j\alpha : \tau_j, \sigma_j \in S^{i-1}, \alpha \in S\}$ through the SGS T;
output Q^i.
end (GIANT_LINK).

8.3. Correctness and time requirement of GIANT_LINK.

THEOREM 8.3. *The outputs T, S^{i-1} of GIANT_LINK are, respectively, an SGS and a set of compatible generators for G_{i-1}/G_i. The collection Q^i is a set of normal generators for G_i.*

Proof. We first claim that after the execution of Step 4, for all $1 \le j \le r$, ρ_j witnesses the separation of j form any j' that is implied by the group $H = \langle Q^{i-1} \cup \{\tau_j, \sigma_j : 1 \le j \le r\}\rangle$. The claim follows from Theorem 6.1 with $G := H, N := H_{B_j}$ and $m := 1$ (because normal generators for the kernel of the action on B_j suffice to witness possible separations of any j' from j). Thus, in particular, the distinct classes $\mathcal{C}_j = \{j' \mid \rho_j|_{B_{j'}} \neq 1\}$, $1 \le j \le r$ partition $\{1, \ldots, r\}$.

When we emerge from Step 5, the distinct classes among the $\mathcal{C}_j$ are again disjoint (a nontrivial intersection would be picked up by the last **if** statement, which would reduce the length of $\rho_{j'}$ for some j' in the intersection) and they now have the same size.

In Step 7, if the sifting of Q^{i-1}, $\sigma_{j'}, \tau_{j'}$ and the elements $w(\tau_j, \sigma_j)$ witness no new separations, then we know that the ρ_j, $1 \le j \le r$, witness all separations implied by elements of the group $H = \langle Q^{i-1} \cup \{\tau_j, \sigma_j : 1 \le j \le r\}\rangle$ (by the argument for Step 4). Furthermore, we know that H acts on L_i as a direct product of alternating groups, exactly one alternating group in each still-linked class of coordinates. If so, the successful sifting of the collection of $\sigma_{j'}^{\alpha}, \tau_{j'}^{\alpha}$ guarantees that H is invariant (mod G_i) under the action of G.

The claims about T and S^{i-1} are now clear. The fact that Q^i is a set of normal generators of G_i then follows from Theorem 6.1 (with the chain of normal subgroups $G = G_0 \ge G_1 \ge \cdots \ge G_{i-1} \ge G_i = N$). □

THEOREM 8.4. *Let $s = |S|$. Then GIANT_LINK$(L_i, Q^{i-1}, S, P_v, \{\alpha_2, \ldots, \alpha_r\}, Q^i, S^{i-1}, T)$ runs in time $O^{\sim}(|Q^{i-1}||L_i|n + s|L_i|^2 n)$ and $|Q^i| \le |Q^{i-1}| + O(s|L_i| + |L_i|^2)$.*

Proof. In Step 1, we can pick an appropriate ρ in $O(|Q^{i-1}|rm)$ and ρ_1 is computed in $O(n)$ time. By Proposition 8.1, Step 2 requires $O(mn)$ time and Step 3 can be executed in $O(rmn)$.

In Step 4, we sift $|Q^{i-1}| + 2r$ elements through r SGS's, requiring $O(|Q^{r-1}|rmn + r^2mn)$ total time; moreover, we compute $O(rm)$ defining relations, in $O(rmn)$ total time. Altogether for all $1 \le j \le r$ we place $O(|Q^{i-1}|r + r^2)$ elements into Σ; for each of these, $O(mr)$ time suffices to check whether it breaks some new links. The total cost of calls of the subroutine GIANT_SEPARATE in Step 4 is at most $O(r^2n)$ since for each pair j, j' we call GIANT_SEPARATE at most once. Hence the total cost of

Step 4 (using that $r \leq n$) is $O(|Q^{r-1}|rmn + r^2mn)$.

We enter the **while** loop of Step 5 at most r times since the minimum length of ρ_j decreases at each call. (Within the **while** loop, the length of each $\rho_{j'}$ decreases at least to the previous minimum.) Hence calls of GIANT_SEPARATE in Step 5 cost $O(r^2n)$. The **if** statement can also be executed within this time bound, since all we have to check is whether the sets $\mathcal{C}_j$ (see the proof of Theorem 8.3) define a partition of $\{1, 2, \ldots, r\}$.

Each time Step 6 is executed, all the ρ_j are of the same length. The length in any round is necessarily a divisor of the length in the previous round, so Step 6 is executed $\leq \log r$ times. By Proposition 8.1, one execution costs $O(mrn)$.

Step 7 is executed always after Step 6, i.e., $\leq \log r$ times, and one execution is similar to Step 4 with the additional sifting of $O(sr)$ conjugates (by S) through the r SGS's for a total timing of $O(|Q^{r-1}|rmn + sr^2mn)$.

Finally, Step 8 runs in $O(|Q^{i-1}|mrn + r^2n + srmn)$. Only the term $O(srmn)$ (instead of $O(sr^2mn)$) requires additional explanation: each conjugate $\alpha^{-1}\tau_j\alpha$, $\alpha \in S$ acts nontrivially in only one of the linked collections of alternating groups so sifting costs only $O(mn)$. Noting that $|L_i| = mr$, the proof is complete. □

Remark 8.5. If $s > n$ then we may use the set $S^* = \bigcup_{j<i-1} S^j$ instead of S as input of GIANT_LINK, replacing the term s by n in both the running time and number of generators created. Correctness is proved by the argument in Remark 7.1.

9. Proof of the main results. In this section, we finish the proof of Theorem 1.1 and sketch two other versions of the algorithm: one with reduced memory requirement (and same time efficiency as the original) and an elementary version with $O^{\sim}(n^{4.5})$ running time.

9.1. Proof of Theorem 1.1. The algorithm described in sections 3–8 computed an SGS for the input group $G = \langle S \rangle \leq \mathrm{Sym}(n)$, $|S| = s$; we have to analyze the running time.

By Lemma 3.1, Corollary 4.7, Proposition 4.8, and Corollary 5.11, the first four steps of the main algorithm run within $O^{\sim}(n^3 + sn^2)$. By the analysis in section 7 and Theorem 8.4, the number of normal generators created while processing level L_{i-1} is $O^{\sim}(s|L_i| + |L_i|^2)$ (in addition to the $|Q^{i-1}|$ normal generators for G_{i-1}). Hence $|Q^i| = O^{\sim}(n^2 + sn)$ for all i and, by section 7 and Theorem 8.4, Step 5 runs in $O^{\sim}(n^4 + sn^3)$.

If the $O^{\sim}(sn^3)$ term becomes dominant, i.e., $s > n$, then we modify the procedure according to Remarks 7.1, 8.5, and the running time drops down to $O^{\sim}(n^4 + sn^2)$. Finally, if sn^2 dominates n^4, i.e., $s > n^2$, then we begin the algorithm by reducing the number of generators to $O(n^2)$ in $O(sn^2)$ time. This can be achieved by sifting the elements of S into (the originally empty) coset representative table with respect to the point stabilizer chain of the permutation domain (cf. procedure SIFT in section 2.7). In any case, we can achieve the claimed $O(n^4 \log^c n + sn^2)$ running time with no logarithmic factors multiplying sn^2.

We turn to the proof of claims (b)–(e) in Theorem 1.1. The order of G is easily computed as the product of sizes of coset representative sets. Although the SGS constructed by the algorithm can be used directly for membership testing by extending the action of a candidate permutation to the SD and there is a method to compute pointwise set stabilizers from it (developed for the parallel procedure in [BLS87]), it is easier to use a result of Brown, Finkelstein, and Purdom [BFP]. They provide an $O(n^3)$ base-change algorithm for converting strong generating sets with respect

to point stabilizer chains along different orderings of the permutation domain. The base-change algorithm outputs the SGS in Jerrum's compact format.

Finally, we observe that the normal sift theorem (Theorem 6.1) essentially provides the scheme for proving (e). Note that, with the descent of the SD complete, the chain generators generate G, so we may assume $S = \bigcup_i S_i$. We associate a symbol x_π to every element π of S and let X denote the collection of these. Each coset representative $\rho \in R_j$ (see notation and discussion preceding the theorem) is representable as a word in S and there is a corresponding word $w(\rho)$ in X. The elements to be sifted in Theorem 6.1 (a),(b),(c) are given as words in S, and so each τ corresponds naturally to a word $w'(\tau)$ in X. Sifting τ can be interpreted as expressing τ canonically as a word $\rho_1 \cdots \rho_l$. From each such sift we derive a relation $w'(\tau)^{-1} w(\rho_1) \cdots w(\rho_l)$ and denote the collection of these by $\mathcal{R}$. Then $\langle X \mid \mathcal{R} \rangle$ is a presentation of G.

9.2. Reducing the memory requirement. The algorithm, as presented in sections 3–8, requires $O^\sim(n^3 + sn^2)$ space. Here we indicate how to reduce the memory requirement to $O^\sim(n^2 + sn)$.

The first four steps of the main algorithm run within this tighter bound. The problem arises because of the top-down approach in Step 5 since, eventually, we accumulate $O^\sim(n^2 + sn)$ normal generators. On the other hand, the SGS we build occupies only $O^\sim(n^2)$ space. The solution is to build the output SGS T simultaneously on all levels. We call T *up-to-date* on level i if $\langle T \cap G_j \rangle$ is a normal subgroup of G for all $j \geq i$. We work always at the lowest level (i.e., greatest index i) which is not up-to-date.

We start executing Step 5 at level 0 as before. The difference is that working on level i, whenever the algorithm produces a normal generator ψ for G_{i+1}, we sift ψ *immediately* into the SGS already constructed. If ψ factors completely then it can be discarded; if it has a nontrivial siftee on some lower level j then we suspend the execution on level i and jump down to level j.

On small levels, we execute exactly the same steps as in the original algorithm (possibly interrupted by some computations on lower levels). On alternating levels, we may execute GIANT_LINK $O(\log n)$ times, discovering smaller and smaller linked collections of subgroups. There are no more than $O(\log n)$ executions since the lengths of linked collections are divisors of each other. This extra work may add a $\log n$ factor to a lower-order term in the running time.

9.3. An elementary version. Two elementary estimates on the order of primitive groups enable us to break the $O(n^5)$ barrier by an elementary, $O^\sim(n^{4.5})$ algorithm. One of them is Pyber's estimate (cf. Theorem 2.5) on the order of nongiant 2-transitive groups and the other one is due to Babai.

THEOREM 9.1 (see [Ba]). *Let $G \leq \mathrm{Sym}(n)$ be primitive; G is not a giant. Then $|G| \leq \exp(O^\sim(\sqrt{n}))$.*

ELEMENTARY ALGORITHM.
INPUT: a set S of generators for $G \leq \mathrm{Sym}(A)$, $|S| = s$.
Step 1. Construct an SF and choose a representative v in each orbit of the SF. For all such v, construct Schreier generators for G_v.
Step 2. For these representatives, use TEST_GIANT to decide whether $G(v)$ is a giant.
By inserting new levels after symmetric levels, obtain the SD. Compute the node stabilizers G_w as in Step 1 for representatives of G-orbits of the SD. Let $(L_0, L_1, \ldots, L_m)$ be the levels of the SD.

Step 3. For each node v representing an alternating level in the SD, construct an SGS for $G(v)$.

Step 4. **for** $i := 1$ to m **do**
if L_{i-1} is an alternating level
then construct SGS for G_{i-1}/G_i, normal generators for G_i as in section 8
else construct SGS for G_{i-1}/G_i, normal generators for G_i as in section 7

end (ELEMENTARY ALGORITHM).

We have to modify the stopping condition in TEST_GIANT and in the first step of THREE_CYCLE to accommodate the weaker bound in Theorem 2.5. This change adds only a logarithmic factor to a low-order term in the running time. Since NATURAL_ACTION is eliminated from this algorithm, correctness is elementary. However, primitive groups on small levels may be of the size allowed in Theorem 9.1, adding a factor $\sqrt{n}$ in the analysis of section 7.

Acknowledgment. We are indebted to the referees for their careful work and for suggestions to improve the presentation.

REFERENCES

[At] M. D. Atkinson, *An algorithm for finding the blocks of a permutation group*, Math. Comp., 29 (1975), pp. 911–913.

[Ba] L. Babai, *On the order of uniprimitive permutation groups*, Ann. of Math., 113 (1981), pp. 553–568.

[BCFLS91] L. Babai, G. Cooperman, L. Finkelstein, E. M. Luks, and Á. Seress, *Fast Monte Carlo algorithms for permutation groups*, in Proc. 23rd ACM Symposium on the Theory of Computing, 1991, pp. 90–100.

[BCFLS95] L. Babai, G. Cooperman, L. Finkelstein, E. M. Luks, and Á. Seress, *Fast Monte Carlo algorithms for permutation groups*, J. Comput. System Sci., 50 (1995), pp. 296–308.

[BLS87] L. Babai, E. M. Luks, and Á. Seress, *Permutation groups in NC*, in Proc. 19th ACM STOC, 1987, pp. 409–420.

[BLS88] L. Babai, E. M. Luks, and Á. Seress, *Fast management of permutation groups*, in Proc. 29th IEEE Foundations of Computer Science, 1988, pp. 272–282.

[BLS93] L. Babai, E. M. Luks, and Á. Seress, *Computing composition series in primitive groups*, in Groups and Computation, DIMACS Series in Discrete Mathematics 11, 1993, pp. 1–15.

[BLS] L. Babai, E. M. Luks, and Á. Seress, *Fast Management of Permutation Groups* II, in preparation.

[BS87] L. Babai and Á. Seress, *On the degree of transitivity of permutation groups: A short proof*, J. Combinatorial Theory Ser. A, 45 (1987), pp. 310–315.

[BS88] L. Babai and Á. Seress, *On the diameter of Cayley graphs of the symmetric group*, J. Combin. Theory Ser. A, 49 (1988), pp. 175–179.

[Bos] R. C. Bose, *Strongly regular graphs, partial geometries, and partially balanced designs*, Pacific J. Math., 13 (1963), pp. 389–419.

[BFP] C. Brown, L. Finkelstein, and P. Purdom, *A new base change algorithm for permutation groups*, SIAM J. Comput., 18 (1989), pp. 1037–1047.

[Cam] P. J. Cameron, *Finite permutation groups and finite simple groups*, Bull. London Math. Soc., 13 (1981), pp. 1–22.

[Car] R. Carter, *Simple Groups of Lie Type*, Wiley, London, 1972.

[CKS] C. W. Curtis, W. M. Kantor, and G. L. Seitz, *The* 2*-transitive permutation representations of the finite Chevalley groups*, Trans. Amer. Math. Soc., 218 (1976), pp. 1–57.

[CM] H. S. M. Coxeter and W. O. J. Moser, *Generators and Relations for Discrete Groups*, 3rd ed., Springer-Verlag, New York, 1972.

[Del] P. Delsarte, *An algebraic approach to the association schemes of coding theory*, Philips Research Report Supplement, 10 (1973).

[FHL] M. L. Furst, J. Hopcroft, and E. M. Luks, *Polynomial time algorithms for permutation groups*, in Proc. 21st IEEE Foundations of Computer Science, 1980, pp. 36–41.

[Go] D. Gorenstein, *Finite Simple Groups and Their Classification*, Academic Press, New York, 1986.

[Ha] M. Hall, Jr., *The Theory of Groups*, Macmillan, New York, 1959.

[HW] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, 5th ed., Clarendon Press, Oxford, 1979.

[Je82] M. Jerrum, *A compact representation for permutation groups*, in Proc. 23rd IEEE Foundations of Computer Science, 1982, pp. 126–133.

[Je86] M. Jerrum, *A compact representation for permutation groups*, J. Algorithms, 7 (1986), pp. 60–78.

[Jo] C. Jordan, *Nouvelles recherches sur la limite de transitivité des groupes qui ne contiennent pas le groupe alterné*, Journ. de Mathématiques, 1 (1895), pp. 35–60.

[Kn] D. E. Knuth, *Efficient representation of perm groups*, Combinatorica, 11 (1991), pp. 33–44.

[Li] M. W. Liebeck, *On minimal degrees and base sizes of primitive groups*, Arch. Math., 43 (1984), pp. 11–15.

[Lu82] E. M. Luks, *Isomorphism of graphs of bounded valence can be tested in polynomial time*, J. Comput. System Sci., 25 (1982), pp. 42–65.

[Lu86] E. M. Luks, *Parallel algorithms for permutation groups and graph isomorphism*, in Proc. 27th IEEE Foundations of Computer Science, 1986, pp. 292–302.

[Lu87] E. M. Luks, *Computing the composition factors of a permutation group in polynomial time*, Combinatorica, 7 (1987), pp. 87–99.

[MS] F. J. Macwilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North–Holland, Amsterdam, 1978.

[Py] L. Pyber, *On the orders of doubly transitive permutation groups, elementary estimates*, J. Combin. Theory Ser. A, 62 (1993), pp. 361–366.

[SS] A. Schönhage and V. Strassen, *Schnelle Multiplikation Großer Zahlen*, Computing, 7 (1971), pp. 281–292.

[Sc] L. L. Scott, *Representations in characteristic p*, in Proc. Santa Cruz Conf. on Finite Groups, AMS, Providence, RI, 1980, pp. 319–322.

[Si67] C. C. Sims, *Graphs and finite permutation groups*, Math. Z., 95 (1967), pp. 76–86.

[Si70] C. C. Sims, *Computational methods in the study of permutation groups*, in Computational Problems in Abstract Algebra, J. Leech, ed., Pergamon Press, Elmsford, NY, 1970, pp. 169–183.

[Wi] H. Wielandt, *Finite Permutation Groups*, Academic Press, New York, 1964.