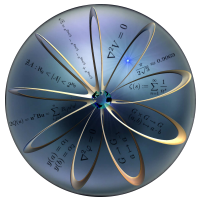

planetmath.org

Math for the people, by the people.

finding the order of a group

Canonical name	FindingTheOrderOfAGroup
Date of creation	2013-03-22 15:54:06
Last modified on	2013-03-22 15:54:06
Owner	Algeboy (12884)
Last modified by	Algeboy (12884)
Numerical id	6
Author	Algeboy (12884)
Entry type	Algorithm
Classification	msc 20B40
Related topic	SchreiersLemma

1 Group order and Membership

Problem 1 (Find Group Order). Given a finite group G, and a subset S of G, determine the order of the subgroup $H = \langle S \rangle$.

Problem 2 (Membership Test). Given a finite group G, and a subset S of G and an element $g \in G$, determine if $g \in H = \langle S \rangle$.

Of course both problems have a simple solution: list all the elements of H by repeated multiplication. However, if $G = S_{100}$, S is some arbitrary subset, it is possible for $\langle S \rangle$ to have order up to 100!, far in excess of any reasonable computation. So the problems are actually feasiblity questions.

At first glance the two problems seem unrelated, but indeed, often they are two versions of the same problem. For example, if we can determine the order of the group $\langle S \rangle$ for any subset S, then given any $g \in G$, compute the order of $\langle S \rangle$ and $\langle S, g \rangle$. If they are equal then $g \in \langle S \rangle$, otherwise $g \notin \langle S \rangle$.

If instead we are able to test membership, then we may sometimes build up to the group H by locating a subgroup of H, testing if the elements of S lie in this subgroup, if not, extend to a larger group building a transversal for the resulting cosets as we go. In the end we have a list of subgroups

$$H = H_k > H_{k-1} > \cdots > H_0 = 1$$

and a set of transversals T_i for H_i/H_{i-1} (as cosets not as quotient groups as H_{i-1} need not be normal). The size of each T_i is the index $[H_i : H_{i-1}]$ and so we can compute the order of H as

$$|H| = [H_k : H_{k-1}] \cdots [H_1 : H_0].$$

Notice this required we build $|T_0| + |T_1| + \cdots + |T_k|$ many elements of H. To make this process efficient requires $|T_i|$ and k be small enough. Part of this is handled in the next basic, yet powerful, result.

Proposition 1. *For every finite group G and every set of generators S of G, there exists a subset of T of S of size no more than $\log_2 |G|$ which also generates G. Furthermore, every chain of subgroups has length no more than $\log_2 |G|$.*

Proof. Given $S = \{s_1, \ldots, s_k\}$ build the subgroup chain $G_i = \langle s_1, \ldots, s_i \rangle$. Notice then that

$$G = G_k \geq G_{k-1} \geq \cdots \geq G_0 = 1.$$

We create T as the subset of $s_{i_r} \in S$ such that $G_{i_r} \neq G_{i_{r-1}}$, that is, we remove elements the generate the same subgroups in the chain. So now

$$G = G_{i_j} > \cdots > G_0 = 1.$$

Hence

$$|G| = [G_{i_j} : G_{i_{j-1}}] \cdots [G_{i_1} : G_0]$$

and $2 \leq [G_{i_r} : G_{i_{r-1}}]$ so $j \leq \log_2 |G|$. □

We now outline the known state of these problems in various computational domains.

2 Permutation Groups

Given $G = S_n$, then both finding the group order and the membership test problem have polynomial times solutions, polynomial in n. The first algorithms of this sort where developed by Charles C. Sims and the computational complexity established by Frust Hopcroft and Luks. These are now known collectively as the Schreier-Sims algorithms because their principle theoretical tool is Schreier's lemma.

To solve use $G^{(i)} = \{g \in G : 1^g = 1, \ldots, i^g = i\}$ which gives a chain of subgroups

$$G = G^{(0)} > G^{(1)} > \cdots > G^{(n-1)} = 1.$$

Applying a careful use of Schreier's lemma to establish transversals of each section of the chain produce the order of G. Moreover, this also implicitly factors every $g \in G$ uniquely into $g_1 g_2 \cdots g_{n-1}$ where $g_i \in G^{(i-1)}$ but $g_i \notin G^{(i)}$. If some $g \in S_n$ cannot be factored through the algorithm (a process usually called *sifting*) then $g \notin G$. Thus memberhsip is also solved.

3 Polycyclic Presentations

A polycyclic presentation by its very design exhibits a chain of subgroups of known prime indices. The order of the group is therefore a produce of the primes. Membership testing can be handled in various ways including sifting the element of the generators of the presentation.

4 Matrix groups

Here the process of computing orders and testing membership collapses to basic problems in number theory: discrete logarithms, and large integer factorization. Unfortunately, these problems have unknown computational complexity and their complexity is the backbone of many cryptographic systems.

For example, given a subgroup H of $GL(2,q)$ for some $q = p^i$, i sufficiently large, it may be that H is a cyclic group, for example a subgroup of the multiplication of the field $GF(q)^\times$. For instance

$$H \leq \left\langle \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \right\rangle.$$

It is a well known hard problem to factor $q-1$ for $q = p^i$ a large power of a prime. Thus it is often the case that we cannot determine subgroups of H as there may be none of small order. Indeed H may be isomorphic to $\mathbb{Z}_r$ for r a very large prime. For example, is $2^{127}-1$ a prime? To determine for a given $g \in GL(2, 2^{127})$ what $|g| = |\langle g \rangle|$ is can require a test of this sort in general.

For membership testing the same number theoretic problems arrise. To test if $g \in H = \langle h \rangle = GF(q)^\times$ would require we find out if $g = h^i$ in $GF(q)$. This is a problem known as the discrete logarithm.

5 General presentations of groups

Given an arbitrary presentation of a group, Boone demonstrates it is impossible even to know if the group is the trivial group. Thus the problem of knowning if the order is non-zero is impossible. Membership testing is therefore also impossible.