Mathematics and Such

Just mathematics – Olympiad or otherwise

Schreier-Sims Algorithm

Posted on June 12, 2018 by limsup

Introduction

Throughout this article, we let *G* be a subgroup of S_n generated by a subset $A \subseteq S_n$. We wish to consider the following questions.

- Given *A*, how do we compute the order of *G*?
- How do we determine if an element $g \in S_n$ lies in *G*?
- Assuming $g \in G$, how do we represent *g* as a product of elements of *A* and their inverses?

In general, the order of *G* is comparable to $|S_n| = n!$ even for moderately-sized *A* so brute force is not a good solution.

We will answer the first two questions in this article. The third is trickier, but there is a nice algorithm by Minkwitz which works for most practical instances.

Application

We can represent the group of transformations of the Rubik's cube as a subgroup of S_{48} generated by a set *S* of 6 permutations. The idea is to label each *non-central* unit square of each face by a number; a transformation of the Rubik's cube then corresponds to a permutation of these 6 × 8 = 48 unit squares.

[Image from official Rubik's cube website.]

Our link above gives the exact order of the Rubik's cube group (43252003274489856000), as computed by the open source algebra package GAP 4. How does it do that, without enumerating all the elements of the group? The answer will be given below.

Schreier-Sims Algorithm

To describe the Schreier-Sims algorithm, we use the following notations:

- $A \subseteq S_n$ is some subset represented in the computer's memory;
- $G = \langle A \rangle$ is a subgroup of S_n;
- *G* acts on the set $[n] := \{1, 2, \ldots, n\}$.

Let us pick some random element $k \in [n]$ and consider its orbit O_k under *G*. From the theory of group actions, we have

$$|O_k| = \frac{|G|}{|G_k|}$$

where $G_k = \{g \in G : g(k) = k\}$ is the isotropy group of *k*. Now it is easy to compute the orbit of *k*: we start by setting the orbit to be the singleton {*k*}, then expand it by letting elements of *A* act on elements of this set. The process stops if we can't add any more elements via this iteration. A more detailed algorithm will be provided later.

Thus, if we could effectively obtain a set of generators for G_k, our task would be complete since we could recursively apply the process to G_k. [Or so it seems: there'll be a slight complication.]

For that, we pick a set *U* of representatives for the left cosets $\{gG_k : g \in G\}$ as follows. For each $j \in O_k$, we pick some element $g \in G_k$ which maps $k \mapsto j$, and for *j* = *k* we pick the identity. To facilitate this process, we use a data structure called a **Schreier vector**.

Schreier Vector for Computing Orbits

Warning: our description of the Schreier vector differs slightly from the usual implementation, since we admit the inverse of a generator.

Let us label the elements of the generating set $A = \{g_1, g_2, \dots, g_m\}$.

- Initialize an array (*v*[1], *v*[2], …, *v*[*n*]) of integers to -1.
- Set *v*[*k*] := 0.
- For each *i* = 1, 2, …, *n*:
  - If *v*[*i*] = -1 we ignore the next step.
  - For each *r* = 1, 2, …, *m*, we set *g* = g_r:
    - set *j* := *g*(*i*); if *v*[*j*] = -1, then set *v*[*j*] = 2*r*-1;
    - set *j* := g^{-1}(*i*): if *v*[*j*] = -1, then set *v*[*j*] = 2*r*.
- Repeat the previous step until no more changes were made to *v*.

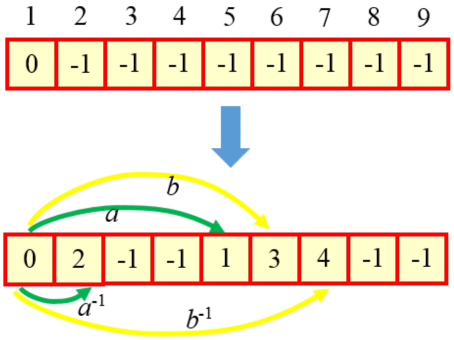
The idea is that *v* contains a "pointer" to an element of *A* (or its inverse) which brings it one step closer to *k*.

Example

Suppose we have elements

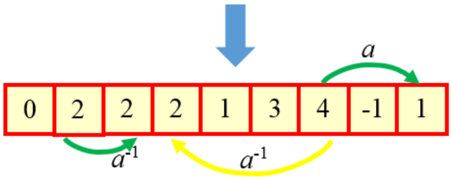
$$a = (1, 5, 3, 2)(4, 7, 9), \quad b = (1, 6, 7).$$

Labelling $a = g_1$ and $b = g_2$, we obtain the following Schreier vector for *k*=1:



1	2	3	4	5	6	7	8	9
0	-1	-1	-1	-1	-1	-1	-1	-1

0	2	-1	-1	1	3	4	-1	-1



0	2	2	2	1	3	4	-1	1

Thus the orbit of 1 is {1, 2, 3, 4, 5, 6, 7, 9}. To compute an element which maps 1 to, say, 9, the vector gives us $ab^{-1}(1) = 9$. Hence we can pick the following for *U*

$$U = \{e, a^{-1}, a^{-2}, a^{-1}b^{-1}, a, b, b^{-1}, ab^{-1}\}.$$

Key Lemma

Next, we define the map $\phi : G \to U$ which takes $g \in G$ to the unique $u \in U$ such that $gG_k = uG_k$ (i.e. *u* is the unique element of *U* satisfying *u*(*k*) = *g*(*k*)). Our main result is:

> ***Schreier's Lemma**.*
>
> *The subgroup G_k is generated by the following set*
>
> $$B := \{\phi(au)^{-1}au : a \in A, u \in U\}.$$

Proof

First note that $\phi(au)^{-1}au$ takes *k* to itself: indeed $\phi(au)$ by definition takes *k* to $au(k)$. Thus we see that each $\phi(au)^{-1}au \in G_k$ as desired so $\langle B \rangle \le G_k$.

Next, observe that *B* is precisely the set of all $u'^{-1}au$ for $u, u' \in U$ and $a \in A$ which lies in G_k. Indeed for such an element, *u'* must be the unique element of *U* which maps *k* to $au(k)$ and so $u = \phi(au)$.

Now suppose $h \in G_k$; we write it as a product of elements of *A* and their inverses:

$$h = a_1^{\epsilon_1} a_2^{\epsilon_2} \dots a_m^{\epsilon_m}, \quad \epsilon_i = \pm 1.$$

We will write

$$u_0^{-1} h u_m = (u_0^{-1} a_1^{\epsilon_1} u_1) \cdot (u_1^{-1} a_2^{\epsilon_2} u_2) \dots (u_{m-1}^{-1} a_m^{\epsilon_m} u_m),$$

where $u_0, u_1, \dots, u_m \in U$ are elements to be recursively chosen. Specifically we start with $u_m := e \in U$, and for each $u_{i+1} \in U$ we set $u_i := \phi(a_{i+1}u_{i+1})$. Note that each term in parentheses is an element of $B \cup B^{-1}$. Thus, the expression lies in G_k.

So we have $u_0 = \phi(hu_m) = \phi(h)$. Since $h \in G_k$, this gives $u_0 = e$ as well so we have obtained *h* as a product of elements of *B* and their inverses. ♦

Example

Consider the subgroup $G \le S_8$ generated by

$$a = (1, 5, 7)(2, 6, 8), \quad b = (1, 5)(3, 4, 8, 2).$$

If we pick *k* = 1, its orbit is $O_k = \{1, 5, 7\}$. For the coset representatives *U*, we take:

$$1 \mapsto g_1 = e, \quad 5 \mapsto g_5 = b, \quad 7 \mapsto g_7 = a^{-1}.$$

Now the subgroup G_k is generated by the following 6 elements:

$$\begin{matrix} g_5^{-1} a g_1 = (2,6,4,3)(5,7), & g_7^{-1} a g_5 = (2,3,4,6)(5,7), & g_1^{-1} a g_7 = e, \\ g_5^{-1} b g_1 = e, & g_1^{-1} b g_5 = (2,4)(3,8), & g_7^{-1} b g_7 = (2,6,3,4)(5,7). \end{matrix}$$

Let $H \leq S_8$ be the subgroup generated by these 6 elements; after removing the identity elements we are left with 4. Now if we pick *k* = 2 next, we obtain 5 representatives for the next *U* and thus, we obtain up to 20 generators for the stabilizer of {1, 2} in *G*.

Problem

The number of generators for the stabilizer groups seems to be ballooning: we started off with 2 examples, then expanded to 6 (but trimmed down to 4), then blew up to 20 after the second iteration.

Indeed, if we naively pick $\{\phi(au)^{-1}au\}$ over all $(a, u) \in A \times U$, then the number of generator increases $|U| = |O_k|$ times, while the order of the group decreases $|O_k|$ times as well. Thus, at worst, the number of generators is comparable to the order of the group, which is unmanageable.

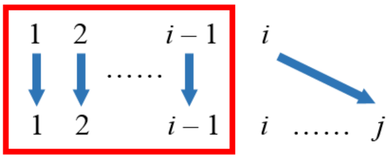
Thankfully, we have a way to pare down the number of generators.

Sims Filter

Sims Filter achieves the following.

> ***Task.***
>
> *Given a set $A \subseteq S_n$, there is an effective algorithm to replace A by some $B \subseteq S_n$ satisfying $\langle A \rangle = \langle B \rangle$ and*
>
> $$|B| \leq \min\{|A|, \tfrac{n(n-1)}{2}\}.$$

Let us explain the filter now. For any non-identity permutation $g \in S_n$, let $J(g)$ be the pair (*i*, *j*) with $1 \leq i < j \leq n$ such that $g(a) = a$ for all $1 \leq a < i$ and $\pi(i) = j$.



Now we will construct a set *B* such that $\langle A \rangle = \langle B \rangle$ and the elements $\pi \in B$ all have distinct $J(\pi)$. It thus follows that $|B| \leq \frac{n(n-1)}{2}$.

1. Label $A = \{g_1, g_2, \ldots, g_m\}$.
2. Prepare a table indexed by (*i*, *j*) for all $1 \leq i < j \leq n$. Initially this table is empty.
3. For each $g_k \in A$, if $g_k = e$ we drop it.
4. Otherwise, consider $J(g_k) = (i, j)$. If the table entry at (*i*, *j*) is empty, we fill g_k in. Otherwise, if the entry is $h \in S_n$, we replace g_k with $g'_k := g_k^{-1}h$ and repeat step 3.
    - Note that this new group element takes *i* to *i* so if it is non-identity, we have $J(g'_k) = (i', j')$ with $i' > i$

After the whole process, the entries in the table give us the new set *B*. Clearly, we have $|B| \leq \min\{|A|, \frac{n(n-1)}{2}\}$.

Example

As above, let us take *A* = {*a*, *b*} with

$$a = (1, 5, 7)(2, 6, 8), b = (1, 5)(3, 4, 8, 2) \in S_8.$$

First step: since *J*(*a*) = (1, 5) we fill the element *a* in the table at (1, 5).

Second step: we also have *J*(*b*) = (1, 5), so now we have to replace *b* with

$$b' := b^{-1}a = (2, 6, 4, 3)(5, 7).$$

Now we have *J*(*b'*) = (2, 6) so this is allowed.

Summary and Applications

In summary, we denote $G^{(0)} = G$ and $A^{(0)} = A$.

- For *i* = 1, 2, …
  - Pick a point $k_i \in [n]$ not picked earlier.
  - Let $G^{(i)}$ be the stabilizer group for k_i under the action of $G^{(i-1)}$. From $A^{(i-1)}$, we use Schreier's lemma to obtain a generating set for $G^{(i)}$.
  - Use Sims filter to reduce this set to obtain $A^{(i)}$ of at most $\frac{n(n-1)}{2}$ elements.
  - If $A^{(i)}$ is empty, quit.

Thus $k_1, k_2, \ldots, k_m \in [n] = \{1, 2, \ldots, n\}$ are distinct such that each of the groups

$$\begin{aligned}&G = G^{(0)} \geq G^{(1)} \geq \ldots \geq G^{(m)} = 1,\\ &G^{(i)} := \{g \in G : g(k_1) = k_1, \ldots, g(k_i) = k_i\}\end{aligned}$$

has an explicit set of generators of at most $\frac{n(n-1)}{2}$ elements. Here are some applications of having this data.

1. Computing |*G*|.

It suffices to compute $\frac{|G^{(i)}|}{|G^{(i+1)}|}$ for each *i*. Since $G^{(i+1)}$ is the stabilizer group for the action of $G^{(i)}$ on k_{i+1} we need to compute the size of the orbit $G^{(i)} \cdot k_{i+1}$ for each *i*. Since we have a generating set for each $G^{(i)}$, this is easy.

2. Determining if a given $g \in S_n$ lies in *G*.

To check if $g \in G$ we first check whether $g(k_1)$ lies in the orbit $G \cdot k_1$. If it weren't, $g \notin G$. Otherwise we pick some $h \in G$ such that $h(k_1) = g(k_1)$. Replacing *g* with $h^{-1}g$, we may thus assume that $g(k_1) = k_1$, and it follows that $g \in G = G^{(0)}$ if and only if $g \in G^{(1)}$. Thus we can solve the problem inductively.

Generalizing, we can determine if $H \leq S_n$ is a subgroup of $G \leq S_n$, when *H* and *G* are given by explicit sets of generators.

3. Checking if $H \leq S_n$ is a normal subgroup of $G \leq S_n$.

Writing $G = \langle A \rangle$ and $H = \langle B \rangle$, we claim that *H* is normal in *G* if and only if:

$$g \in A, h \in B \implies ghg^{-1} \in H.$$

First we fix $g \in G.$ Since $(ghg^{-1})(gh'g^{-1}) = ghh'g^{-1}$ and $(ghg^{-1})^{-1} = gh^{-1}g^{-1}$, we see that $gHg^{-1} \subseteq H.$ But both groups are of the same *finite* cardinality so equality holds. Thus $g^{-1}Hg = H$ as well. It follows that $gHg^{-1} = H$ for all $g \in G.$

Share this:

Twitter Facebook

Like

Be the first to like this.

Related

Solving Permutation-Based Puzzles
June 21, 2018
In "group actions"

Casual Introduction to Group Theory (1)
September 21, 2012
In "Notes"

Casual Introduction to Group Theory (3)
September 25, 2012
In "Notes"

This entry was posted in Uncategorized and tagged group actions, group theory, permutations, programming, rubik's cube, schreier-sims, symmetries. Bookmark the permalink.

Mathematics and Such

Blog at WordPress.com.