

CS299 Mid-Sem Report

Ruchit Jagodara (22110102) (ruchit.jagodara@iitgn.ac.in)

Pranav Joshi (22110197) (pranav.joshi@iitgn.ac.in)

Problem Description

Implementation of the first polynomial time (in the cardinality of Group) Minimum Generating Set algorithm based on Dhara Thakkar and Andrea Lucchini's [research paper](#) in [SageMath](#) and getting it accepted in SageMath via a Pull Request to Sage's [GitHub repository](#).

Preliminary Literature Survey

To understand the algorithm in detail, we read about these topics:

1. Normal Groups, Quotient Groups, Isomorphism, Homomorphism, Simple Groups, Solvable Groups, Normal Closure.
2. Various cases of Groups, like Cyclic groups, Permutation Groups, Special Linear Groups, General Linear Groups.

We talked with one of the authors, [Dhara Thakkar](#), of the Research Paper and understood the Algorithm and we learned the inner workings and usage of SageMath and [GAP](#) via Sage by going through the codebase of SageMath.

Work

First, we formed basic pseudo-code.

We searched for many functions required for the implementation. Although we were not able to find every function needed, we also enquired from the SageMath community. However, the functions they initially suggested worked differently from what we wanted.

So, we implemented our [own library](#) for Finite Groups based on Cayley Table representation, which is what the algorithm was using. We implemented various functionalities like Cosets, Conjugates, Quotient Group, Minimal Normal Subgroup (via normal closures), cyclic generator, Minimal Generating Set, etc., in our custom library, and finally, Minimum Generating Set and [tested](#) our implementation on various finite groups.

We wanted to add this whole library to Sage, but many functions we wrote were taking a lot of time compared to Sage's implementation (which is highly optimized). Nevertheless, our custom library served its purpose. We talked with the author again to ensure that our implementation was working correctly.

Meanwhile, we were constantly asking around in the Sage community for help so that we could complete the old (Sage) implementation, and luckily, we found what we needed, so we continued with the old implementation because it was using highly optimized functions from Sage and GAP.

We then timed the function and found that it was indeed polynomial time in $|G|$. We found this by fitting logarithmic curves to the plot of $\ln(t)$ vs G , where t is the time taken from the algorithm in seconds.

You can see all the results [here](#).

Then we opened an issue in Sage for adding the feature for minimum generating set.