WXML Final Report: Randomness of the Discrete Logarithm

Jayadev Athreya, Chris Hoffman, Jacob Richey, Kristine Hampton, . . .

Spring 2016

1 Introduction

1.1 The initial problem

The initial question we investigated was whether or not the discrete logarithm behaves randomly. The discrete logarithm is the integer solution, k, to the equation $a^k \equiv b \bmod n$. We notate this by $dlog_a b \equiv k \bmod n$. We were particularly interested in the situation in which n is a prime p. In this case, the set of all possible values the discrete logarithm can take is contained within the cyclic abelian group $\mathbb{Z}_p^* \cong \mathbb{Z}_{p-1}$. The bases of the discrete log we are interested in, a, are called primitive roots and are exactly the generators of $\mathbb{Z}_p^*$. Each discrete logarithm with a primitive root base defines a permutation of the integers $\{1, 2, ..., p-1\}$.

1.2 New Directions

We began by comparing the behavior of the discrete logarithm with that of a random permutation, to attempt to determine if the discrete logarithm permutation behaves in a random manner. After gathering observational evidence in this manner, we started to question if this was the appropriate way of quantifying randomness. Our method was highly subjective, relying on our opinion if the discrete log permutation statistics "looked" reasonably similar to the random permutation statistics. As such, our problem split into a three different directions: trying to determine what it means to be a random mathematical object, trying to determine which aspects of the discrete logarithm behaved similar to a random permutation and trying to prove which aspects of the discrete logarithm are highly deterministic. As we branched outside of our initial question, instead of viewing the discrete logarithm as only a permutation, we began to view it as an isomorphism $\sigma : \mathbb{Z}_p^* \to \mathbb{Z}_{p-1}$, an object that was easier for us to intuitively understand.

2 Progress

2.1 Computational

Throughout our project, we wrote code to generate a variety of statistics on the discrete logarithm. Given any prime p, there is code to determine the primitive roots, the transition maps between primitive roots, the cycle notation and structures of each discrete logarithm map and various statistics for each permutation such as number of fixed points and longest cycle length. There is also code to graphically show where each discrete logarithm for any prime sends any integer k and the frequency that each point is an image for k under the discrete logarithm map.

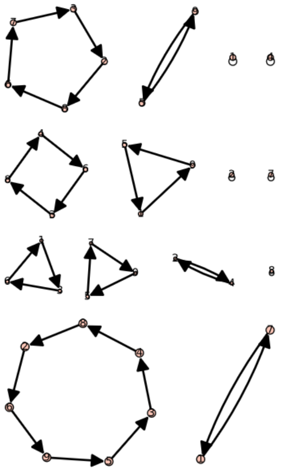
2.2 Theoretical

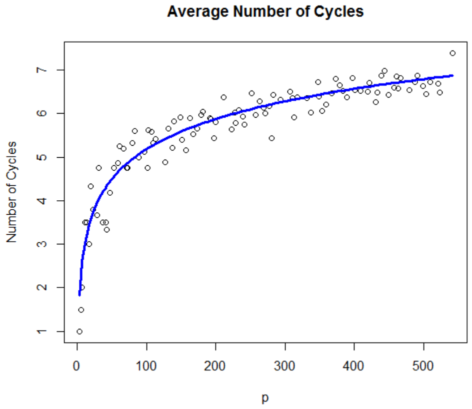
During the project we came up with two conjectures. We were able to prove our first conjecture, which is that the image of the discrete logarithm is parity sensitive. Formally, for any prime p and integer k, if $dlog_a k$ mod p is even for some primitive root a then it is even for all primitive roots. Conversely, if it is odd for some primitive root then it is odd for all primitive roots. The second conjecture we are attempting to prove deals with the uniqueness of the images of any element under the discrete logarithm map. Formally it states that for all $k \in \mathbb{Z}_p^*$, if $k^d \equiv 1$ mod p for some integer d then the number of primitive roots that map k to the same number is $\phi(p-1)/\phi(d)$.

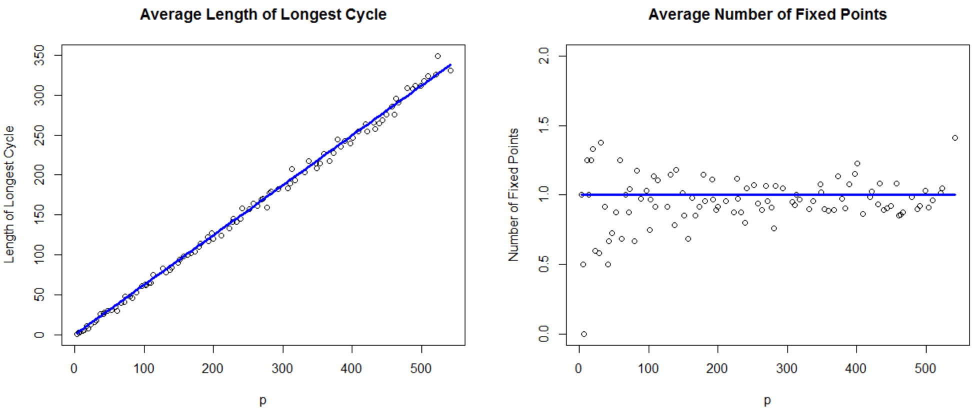
3 Future Directions

Computationally, as we move forward our code needs to be updated to be more efficient. Currently, the code begins to slow down drastically when the prime we are modding out by gets too large. Theoretically, the overarching problem in our research has been determining what it means for a deterministic object to be random. To progress on our initial question, we will need to come up with a way of quantifying randomness. In addition to this, we are going to continue to work on quantifying the randomness of the discrete logarithm and proving our second conjecture. As the discrete logarithm is not limited to the set of integers modulo a prime, we would also like to expand the problem we are working on and work outside of finite integer fields, particularly in groups of matrices. Moving forward we will continue to work on our initial question, trying to determine what aspects of the discrete logarithm is random, but also look at new environments for the discrete logarithm to operate in and see how our results change in these new environments.

4 Code Output






Average Length of Longest Cycle
Length of Longest Cycle
0
50
100
150
200
250
300
350
0
100
200
300
400
500
p
Average Number of Fixed Points
Number of Fixed Points
0.0
0.5
1.0
1.5
2.0
0
100
200
300
400
500
p