Pollard's Rho Algorithm

Jazlynn Leung

April 2024

Introduction

Pollard's Rho algorithm is a method for determining a factor of a composite number. The algorithm makes use of the "birthday paradox" and the "tortoise and hare" concept.

Outline

Given a composite number n, we want to iteratively generate k "random" numbers $t_1, t_2, \cdots, t_k$ by using a "randomizer" function $g(t)$. We usually use $g(t) = t^2 + c$, for some number c. We'll use $c = 1$ to demonstrate.

We start by selecting a starting value t_0. We'll use $t_0 = 2$ to demonstrate. This is our "tortoise". We recursively define $t_i = g(t_{i-1})$ for $i \geq 1$.

We then define $h_0 = t_0$. This is our "hare". We recursively define $h_i = g(g(h_{i-1}))$ for $i \geq 1$. Notice that $h_i = t_{2i}$ for each i.

Iteration

We now iterate along the sequences t_i and h_i until a certain condition is satisfied. In particular, we calculate $d = \gcd(h_i - t_i, n)$ on the i-th iteration using the Euclidean algorithm.

Case 1: $d = 1$:
We have gained no information, so we increment i and try again.

Case 2: $d = n$: We have gained no information, and we never will using these c and t_0 values. We terminate the algorithm and retry with new c and/or t_0 values.

Case 3: $1 < d < n$: We have found a non-trivial factor d of n, and the algorithm terminates successfully.

Tortoise and Hare

The tortoise and hare concept is the basis of Floyd's cycle detection algorithm, which searches for a loop in the sequence of t_i's when taken mod d. One can show that such a loop always exists mod d, and that there must be some i such that $h_i - t_i = t_{2i} - t_i$ is divisible by d. The existence of the loop gives rise to the Rho shape. In most cases, $h_i - t_i$ will *not* be divisible by n, and thus $\gcd(h_i - t_i, n)$ is some proper divisor of n (usually d, but maybe a multiple of d). This corresponds to case 2. If $h_i - t_i$ *is* divisible by n, we get no useful information, corresponding to case 3.

The Birthday Paradox

In a room of 23 people, there are $23!\binom{365}{23}$ ways in which each could have a distinct birthday. There are 365^{23} possible ways in which they could have not necessarily distinct birthdays. Thus, the chance that some pair of them share a birthday is

$$1 - \frac{23!\binom{365}{23}}{365^{23}} > 50\%$$

In other words, it is *more likely than not* that some pair of people share a birthday out of just 23 people.

The more general mathematical fact is that about $\sqrt{n}$ selections from n items are required to get a better than even chance of selecting some item twice.

The Birthday Paradox

The birthday paradox applies to the Pollard rho algorithm by estimating how quickly the sequence of t_i's will loop. Mod d, there are d possibilities for each t_i, so after about $\sqrt{d}$ terms are computed, there is a better than even chance that two of them repeat, leading to a loop.

Thank you for listening!