5.6.2 The Pollard Rho Algorithm

Let p be a prime divisor of n. Suppose there exist two integers $x, x' \in \mathbb{Z}_n$, such that $x \neq x'$ and $x_i \equiv x_j \pmod{p}$. Then $p \leq \gcd(x - x', n) < n$, so we obtain a non-trivial factor of n by computing a greatest common divisor. (Note that the value of p does not need to be known ahead of time in order for this method to work.)

Suppose we try to factor n by first choosing a random subset $X \subseteq \mathbb{Z}_n$, and then computing $\gcd(x - x', n)$ for all distinct values $x, x' \in X$. This method will be successful if and only if the mapping $x \mapsto x \bmod p$ yields at least one collision for $x \in X$. This situation can be analyzed using the birthday paradox described in Section **??**: if $|X| \approx 1.17\sqrt{p}$, then there is a 50% probability that there is at least one collision, and hence a non-trivial factor of n will be found. However, in order to find a collision of the form $x \bmod p = x' \bmod p$, we need to compute $\gcd(x - x', n)$. (We cannot explicitly compute the values $x \bmod p$ for $x \in X$, and sort the resulting list, as suggested in Section **??**, because the value of p is not known.) This means that we would expect to compute more than $\binom{|X|}{2} > p/2$ $\gcd$'s before finding a factor of n.

The POLLARD RHO ALGORITHM incorporates a variation of this technique that requires fewer $\gcd$ computations and less memory. Suppose that the function f is a polynomial with integer coefficients, e.g., $f(x) = x^2 + a$, where a is a small constant ($a = 1$ is a commonly used value). Let's assume that the mapping $x \mapsto f(x) \bmod p$ behaves like a random mapping. (It is of course not "random", which means that the analysis we are presenting is not a rigorous proof.) Let $x_1 \in \mathbb{Z}_n$, and consider the sequence $x_1, x_2, \ldots$, where

$$x_j = f(x_{j-1}) \bmod n,$$

for all $j \geq 1$. Let m be an integer, and define $X = \{x_1, \ldots, x_m\}$. To simplify matters, suppose that X consists of m distinct residues modulo n. Hopefully it will be the case that X is a random subset of m elements of $\mathbb{Z}_n$.

We are looking for two distinct values $x_i, x_j \in X$ such that $\gcd(x_j - x_i, n) > 1$. Each time we compute a new term x_j in the sequence, we could compute $\gcd(x_j - x_i, n)$ for all $i < j$. However, it turns out that we can reduce the number of $\gcd$ computations greatly. We describe how this can be done.

Suppose that $x_i \equiv x_j \pmod{p}$. Using the fact that f is a polynomial with integer coefficients, we have that $f(x_i) \equiv f(x_j) \pmod{p}$. Recall that $x_{i+1} = f(x_i) \bmod n$ and $x_{j+1} = f(x_j) \bmod n$. Then

$$x_{i+1} \bmod p = (f(x_i) \bmod n) \bmod p = f(x_i) \bmod p,$$

because $p \mid n$. Similarly,

$$x_{j+1} \bmod p = f(x_j) \bmod p.$$

Therefore, $x_{i+1} \equiv x_{j+1} \pmod p$. Repeating this argument, we obtain the following important result: if $x_i \equiv x_j \pmod p$, then $x_{i+\delta} \equiv x_{j+\delta} \pmod p$ for all integers $\delta \geq 0$. Denoting $\ell = j - i$, it follows that $x_{i'} \equiv x_{j'} \pmod p$ if $j' > i' \geq i$ and $j' - i' \equiv 0 \pmod \ell$.

Suppose that we construct a graph G on vertex set $\mathbb{Z}_p$, where for all $i \geq 1$, we have a directed edge from x_i to x_{i+1}. There must exist a first pair x_i, x_j with $i < j$ such that $x_i = x_j$. By the observation made above, it is easily seen that the graph G consists of a "tail" $x_1\, x_2\, \ldots\, x_i$, and an infinitely repeated cycle of length ℓ, having vertices $x_i\, x_{i+1}\, \ldots\, x_j = x_i$. Thus G looks like the greek letter ρ, which is the reason for the name "rho algorithm".

We have already mentioned that we want to discover two terms $x_i = x_j$ with $i < j$, by computing a $\gcd$. It is not necessary that we discover the first occurrence of a collision of this type. In order to simplify and improve the algorithm, we restrict our search for collisions by taking $j = 2i$. The resulting algorithm is presented as Algorithm 5.9.

Algorithm 5.9: POLLARD RHO FACTORING ALGORITHM(n, x_1)

external f
$x \leftarrow x_1$
$x' \leftarrow f(x) \bmod n$
$p \leftarrow \gcd(x - x', n)$
while $(p = 1)$ **or** $(p = n)$

$$\textbf{do} \begin{cases} \textbf{comment: } \text{in the } i\text{th iteration, } x = x_i \text{ and } x' = x_{2i} \\ x \leftarrow f(x) \bmod n \\ x' \leftarrow f(x') \bmod n \\ x' \leftarrow f(x') \bmod n \\ p \leftarrow \gcd(x - x', n) \end{cases}$$

return (p)

This algorithm is not hard to analyze. If $x_i = x_j$, then it is also the case that $x_{i'} = x_{2i'}$ for all i' such that $i' \equiv 0 \pmod \ell$ and $i' \geq i$. Among the ℓ consecutive integers $i, \ldots, j-1$, there must be one that is divisible by ℓ. Therefore the smallest value i' that satisfies the two conditions above is at most $j - 1$. Hence, the number of iterations required to find a factor p is at most j, which is expected to be at most $\sqrt{p}$. Since $p < \sqrt{n}$, the expected complexity of the algorithm is $O(n^{1/4})$, ignoring logarithmic factors. However, we again emphasize that this is a heuristic analysis, and not a mathematical proof. On the other hand, the actual performance of the algorithm in practice is similar to this estimate.

When Algorithm 5.9 is run with $n = 15770708441$ (the same value of n considered in Example 5.9), $x_1 = 1$ and $f(x) = x^2 + 1$, it is discovered that $x_{422} = 2261992698$, $x_{211} = 7149213937$, and

$$\gcd(x_{422} - x_{211}, n) = 135979.$$