Tonelli-Shanks Algorithm and Berlekamp's Algorithm

Notes by Linus Tang.

These notes have not been thoroughly reviewed. Any errors below are my own responsibility.

Sources:
- Folklore / Wikipedia editors
  - https://en.wikipedia.org/wiki/Tonelli%E2%80%93Shanks_algorithm
  - https://en.wikipedia.org/wiki/Berlekamp%27s_algorithm

Problem

Find an efficient (polylog(p)) algorithm that, given integer n and prime p, determines whether n is a quadratic residue in $\mathbb{F}_p$ and if so, finds a squareroot.

Trivial progress

Euler's criterion: n is a quadratic residue if and only if $n^{\frac{p-1}{2}} = 1$.

If $p \equiv 3 (\mathrm{mod}\, 4)$, we can simply take $r = \pm n^{\frac{p+1}{4}}$ and have $r^2 = n^{\frac{p+1}{2}} = n$. But the case $p \equiv 1 (\mathrm{mod}\, 4)$ is nontrivial.

Solution (Tonelli-Shanks algorithm)

Assume $x \neq 0$, since that case is trivial.

We use the fact that $\mathbb{F}_p^\times$ is cyclic. We can write $p - 1 = Q \cdot 2^S$ for odd Q and a positive integer S. One intuition is that $\mathbb{F}_p^\times$ is isomorphic to $\mathbb{Z}_Q \times \mathbb{Z}_{2^S}$, and as we'll see later, it's easy to find an element in the correct $\mathbb{Z}_{2^S}$-coset, but after that locating the exact correct element within the coset takes a little care.

Algorithm, informal description

- Verify that $n^{\frac{p-1}{2}} = 1$, as per Euler's criterion.
- Write $p - 1 = Q \cdot 2^S$ with Q odd.
- Set $R = n^{\frac{Q+1}{2}}$ so that $R = n(n^Q) = nt$.

Notice that $t^{2^{S-1}} = 1$ because the left side is $n^{Q \cdot 2^{S-1}} = n^{\frac{p-1}{2}} = 1$. However, we would like to have $t^{2^0} = 1$ (i.e. $t = 1$), which would let us have $R^2 = nt = n$.

This is what was meant earlier by "finding an element in the correct $\mathbb{Z}_{2^S}$-coset". We will proceed to find the exact correct element by adjusting R and t.

We adjust R and t, keeping the invariant that $R^2 = nt$, and that $\mathrm{ord}_p(t)$ is a power of 2 (we stay in the correct $\mathbb{Z}_{2^t}$-coset). We do this in such a way that decreases $\mathrm{ord}_p(t)$ until it becomes 1. To keep the first invariant, our updates should be of the form

- $r \leftarrow br$
- $t \leftarrow b^2 t$

To keep the second invariant, we should have that $\mathrm{ord}_p(b)$ is a power of 2.

Recall that we have $\mathrm{ord}_p(t) = 2^m$ for some $m \leq S - 1$. It turns out that if we choose b such that $\mathrm{ord}_p(b^2) = 2^m$, then $\mathrm{ord}_p(tb^2) = 2^{m'}$ for some $m' < m$. (This is similar to the fact that if $\nu_2(x) = \nu_2(y) = -m$, then $\nu_2(x + y) = -m'$ for some $m' < m$.)

How do we find such b? We can take any quadratic non-residue z (since about half of all elements are non-residues, this can be done efficiently by searching randomly), and notice that $\mathrm{ord}_p(z^Q) = 2^S$, so taking $b = z^{Q \cdot 2^{S-m-1}}$ works.

Thus, our algorithm is as follows.

Algorithm, formal description

- Verify that $n^{\frac{p-1}{2}} = 1$, as per Euler's criterion.
- Find a quadratic nonresidue $z \in \mathbb{F}_p$.
- Write $p - 1 = Q \cdot 2^S$ with Q odd.
- Set $R = n^{\frac{Q+1}{2}}$ so that $R = n\big(n^Q\big) = nt$.
- While $t \neq 1$:
  - Compute $\operatorname{ord}_p(t)$, which is a power of 2; let it be 2^m.
  - Let $b = z^{Q \cdot 2^{S-m-1}}$.
  - Update $r \leftarrow br$.
  - Update $t \leftarrow b^2 t$.

As explained above, m decreases with each loop, so t eventually equals 1. Since the invariant $R^2 = nt$ is preserved, we will have $R^2 = n$, so the square roots of n are $\pm R$.

Generalization to higher order roots

We can easily generalize Tonelli-Shanks to find the k-th root of a given k-th power residue $n \in \mathbb{F}_p$, for small k.

Factoring squarefree polynomials over finite fields (Berlekamp's algorithm)

Indeed, it is possible to efficiently find the factorization any polynomial $f(x)$ over any finite field $\mathbb{F}_q$ (with $q = p^r$).

We first solve the squarefree case. Consider squarefree $f(x) \in \mathbb{F}_q[x]$, which factorizes into irreudcible polynomials as $f(x) = f_1(x) \cdots f_k(x)$.

Since f is squarefree, $f_1, ..., f_k$ are nonassociate, so there is an isomorphism

$$\sigma : \mathbb{F}_q[x]/(f(x)) \to \prod_{i=1}^{k} \mathbb{F}_q[x]/(f_i(x))$$

given by Chinese remainder theorem. Consider the map $h(x) \mapsto h(x^p) = h(x)^p$, which trivially commutes with σ.

Then σ restricts to an isomorphism

$$\sigma : \operatorname{Fix}_p\big(\mathbb{F}_q[x]/(f(x))\big) \to \prod_{i=1}^{k} \operatorname{Fix}_p\big(\mathbb{F}_q[x]/(f_i(x))\big)$$

between the fixed fields. Here, $\operatorname{Fix}_p(R)$ denotes the fixed field of R under $h(x) \mapsto h(x)^p$, i.e.

$$\operatorname{Fix}_p(R) = \{h(x) \in R : h(x) = h(x)^p\}.$$

Note that for all $1 \leq i \leq k$, since f_i is irreducible, $\mathbb{F}_q[x]/(f_i(x))$ is a field of characteristic p. By a classic result in finite field theory, the fixed field of $h(x)$ consists of the constant polynomials and is isomorphic to $\mathbb{F}_p$.

Therefore, if we can determine $A = \operatorname{Fix}_p\big(\mathbb{F}_q[x]/(f(x))\big)$, we immediately know whether f is irreducible. In particular, A is isomorphic to $\prod_{i=1}^{k} \mathbb{F}_p$.

Furthermore, note that for any $g(x) \in A$, working in $\mathbb{F}_q[x]/(f(x))$, we have

$$g(x)^{\frac{p-1}{2}} = \begin{cases} 0 & g = 0 \\ 1 & g \text{ is a nonzero square} \\ -1 & g \text{ is not a square} \end{cases}$$

which means that a random $g \in A$ will satisfy $f_i | g^{\frac{p-1}{2}} - 1$ with probability about $\frac{1}{2}$ (independently over i by Chinese remainder theorem).

Therefore, once we compute A, if its dimension is $k > 1$ as a $\mathbb{F}_p$-vector space, we can repeatedly sample a random $g(x) \in A$ and compute $\gcd\left(f(x), g(x)^{\frac{p-1}{2}}\right)$, which will be a nontrivial factor of f with probability about $1 - \frac{1}{2^{k-1}}$.

Computing A is a matter of writing a matrix representing the linear relations in the coefficients of $h(x)$ which are necessary and sufficient for $h(x) = h(x^p)$, then putting it in reduced row echelon form to read off the nullspace.

So, we have an efficient algorithm, which given a squarefree polynomial $f(x)$ over $\mathbb{F}_q$, either determines that it is irreducible or finds a nontrivial factor. We can repeatedly apply this algorithm to each factor until f is completely reduced.

In order to handle the general case ($f(x)$ not necessarily squarefree), note that the factors that divide $f(x)$ with multiplicity exactly ℓ divide the first $\ell - 1$ derivatives of f (but not the ℓ-th derivative). Thus, we can decompose f into

$$\frac{f}{\gcd(f, f')} \cdot \frac{\gcd(f, f')}{\gcd(f, f', f'')} \cdot \frac{\gcd(f, f', f'')}{\gcd(f, f', f'', f''')} \cdots,$$

each of which is squarefree, and reduce each factor above into irreducibles.