0.1 Square roots mod p (Tonelli-Shanks)

- *Algorithm:* Tonelli–Shanks (algo. 1)
- *Input:* a prime number p and a positive integer n in $\mathbb{Z}/p\mathbb{Z}$.
- *Complexity:* $\mathcal{O}(n^2)$
- *Data structure compatibility:* N/A
- *Common applications:* cryptography research, number theory research

Problem. Square roots mod p (Tonelli-Shanks)

Given a prime number p and a positive integer n in $\mathbb{Z}/p\mathbb{Z}$, we want to find the quadratic residue of n, which is a positive integer r in $\mathbb{Z}/p\mathbb{Z}$, such that $r^2 \equiv n \mod p$.

Description

When designing an algorithm to solve for a computer science problem, we would like to seek for a fast algorithm. If we can get a polynomial time complexity, we will be satisfied and the implementation is workable. But in cryptography study, we do the reverse. The security requires a significantly high level of time complexity if someone else want to recover the secret. However, we should notice that although it should be designed to be hard to recover the secret, using the secret(its inverse) should be fast to process.

In the field of number theory, there is an interesting question that given a large prime number p and a positive interger n in $\mathbb{Z}/p\mathbb{Z}$, we would like to find out an integer r in $\mathbb{Z}/p\mathbb{Z}$ such that $r^2 \equiv n \mod p$. This is also called as finding a quadratic residue. Notice that these calculations are done under modular arithmetic within the field $\mathbb{Z}/p\mathbb{Z}$.

Further, Tonellis's algorithm would be extended for any cyclic group and to the kth root instead of square root. A table can be prepared in advance to facilitate the process. Another extension would be this algorithm can be used on module of p^k instead of p. For this extension, the proof and process is more complicated and can be seen in Dickson's *Theory of Numbers*[1] and is beyond the scope of this course[4]. The usage of this algorithm is mostly in the field of cryptography research like Rabin cryptography system and elliptic curves.

Time complexity analysis is very difficult for this algorithm. According to the paper written by Lindhurst and Scott. The average case time complexity would be $\frac{n^2}{4} + \frac{7n}{4} + 1$ multiplications with the standard deviation of deviation $\sqrt{\frac{n^3}{12} + \frac{3n^2}{8} + \frac{13n}{24} - 1}$[2]. The worst case is easier to analysis, which should be $(n+2)+(n+1)+...+5+4 = \frac{1}{2}\left(n^2 + 5n - 6\right)$ multiplications. Basically, the algorithm time complexity indicates that it is easy for eve to recover the secret. So Alice and Bob should never use a simple prime number p for quadratic residue. Instead, they should use $p \times q$ instead of p, where p and q are both large prime numbers.

References.

[1] Leonard Eugene Dickson. *History of the Theory of Numbers*. Carnegie Institution of Washington, 1919 (cit. on p. 1).

[2] Scott Lindhurst. "An analysis of Shanks's algorithm for computing square roots in finite fields". In: *To appear in the proceedings of the* (1997) (cit. on p. 1).

Algorithm 1: Tonelli–Shanks

Input : a prime number p and a positive integer n in $\mathbb{Z}/p\mathbb{Z}$
Output: a integer r in $\mathbb{Z}/p\mathbb{Z}$ such that $r^2 \equiv n \mod p$.

Factorize the number p - 1 by factor 2 $Q \leftarrow p-1$ $s \leftarrow 0$ **while** $2 \mid Q$ **do**
    $Q \leftarrow Q/2$
    s++
end while
We now get $p - 1 = 2^s \times Q$.
Use Jacobi symbol to rule out the elements in $\mathbb{Z}/p\mathbb{Z}$ that are quadratic non-residues.
$M \leftarrow S$
$c \leftarrow z^Q$
$t \leftarrow n^Q$
$R \leftarrow n^{\frac{Q+1}{2}}$
while $t \neq 0$ **and** $t \neq 1$ **do**
    find the least i, $0 < i < M$, such that $t^{2^i} = 1$.
    $b \leftarrow c^{2^{M-i-1}}$
    $M \leftarrow i$
    $c \leftarrow b^2$
    $t \leftarrow tb^2$
    $R \leftarrow Rb$
end while
if $t = 0$ **then**
    return r = 0
end if
if $t = R$ **then**
    return r = R
end if
/* notice that if we have the answer r, it is obvious that $-r$(or $p - r$) is another answer for $r^2 \equiv n \mod p$.[3] */

[3] Manuel. *VE475 – Introduction to Cryptography (lecture slides)*. 2020 (cit. on p. 2).

[4] Wikipedia contributors. *Tonelli–Shanks algorithm*. [Online; accessed 6-October-2020]. 2020. URL: #https://en.wikipedia.org/wiki/Tonelli%E2%80%93Shanks_algorithm#cite_note-dickson-3 (cit. on p. 1).