Generators of Z_p^*

Rikard Hjort
hjortr@student.chalmers.se

July 23, 2018

1 Generators of Z_{17}^*

Since $\phi(16) = 8$ we expect to find 8 generators. The table below shows which they are. The top column lists the generators, and each row represents taking the generator to the power $2, 3, 4, ...16$.

a	3	5	6	7	10	11	12	14
a^2	9	8	2	15	15	2	8	9
a^3	10	6	12	3	14	5	11	7
a^4	13	13	4	4	4	4	13	13
a^5	5	14	7	11	6	10	3	12
a^6	15	2	8	9	9	8	2	15
a^7	11	10	14	12	5	3	7	6
a^8	16	16	16	16	16	16	16	16
a^9	14	12	11	10	7	6	5	3
a^{10}	8	9	15	2	2	15	9	8
a^{11}	7	11	5	14	3	12	6	10
a^{12}	4	4	13	13	13	13	4	4
a^{13}	12	3	10	6	11	7	14	5
a^{14}	2	15	9	8	8	9	15	2
a^{15}	6	7	3	5	12	14	10	11
a^{16}	1	1	1	1	1	1	1	1

These numbers seems correct on inspection. For example, $a^8 = 16 = -1$ for all generators, which is in coordance with Proposition 5, and $a^9 = -a$, accordingly.

2 Prove propositions

2.1 If $k \mid p-1$ then $|O_k| = \phi(k)$

Proof: Any multiplication group modulo n is cyclic if n is a prime (generally, if it is prime to any positive power). Since Z_p^* is thus cyclic, it has at least one generating element. Let a be any generator of Z_p^*, and $1 \leq d \leq p-1$.

$\gcd(d, p-1) \mid d$ and $\gcd(d, p-1) \mid p-1$ which means that if we set $k = (p-1)/\gcd(d, p-1)$ then $dk = m(p-1)$ for $m = d/\gcd(d, p-1)$. By Fermat's little theorem, $a^{dk} = a^{m(p-1)} = 1$. Since $(a^d)^k = 1$, the order of a^d is some divisor of k. But by our choice of k it contains only the prime factors not common to d and $p-1$. Dividing by any of these prime factors (the only valid division of k), obtaining k' would result in a number for which $p-1 \nmid dk'$, since there is some prime factor in $p-1$ missing in dk'. Thus, the smallest possible number we can multiply d with to obtain a multiple of $p-1$ is k, which means the order of a^d is exactly k. Now, we will examine how many possible choices there are of d for which k remains unchanged. Set $d = fk, 1 \leq f \leq (p-1)/k = \gcd(d, p-1)$. The choices of f for which k remains unchanged are exactly those which do not contain any of the prime factors of k, or $\gcd(f, k) = 1$. The number of such choices are exactly $\phi(k)$ by definition. Thus, we have proven that the number of elements a^d of order k is $\phi(k)$, *Q.E.D.*.

2.2 $\forall n > 1, \frac{\phi(n)}{n} = \Omega\left(\frac{1}{\log n}\right)$

We begin by observing that

$$\phi(n) = \prod_{(p,k)\in P(n)} p^k\left(1 - \frac{1}{p}\right) = n \prod_{(p,k)\in P(n)} 1 - \frac{1}{p}$$

where $P(n) = \{(p, k) \mid p^k \text{ is in the prime factorization of } n\}$. This in turn means that

$$\frac{\phi(n)}{n} = \prod_{(p,\_)\in P(n)} 1 - \frac{1}{p}$$

Let p_i be any prime dividing n. We know that $p_i \geq 2$, and thus $1 - \frac{1}{p} \geq \frac{1}{2}$. This also means that $|P(n)|$ can not be larger than $\log_2 n$, which replacing each p_i with 2 in

$$p_1 p_2 ... p_{|P(n)|} \leq n = 2^{\log_2 n}$$

should make evident. Thus,

$$\frac{\phi(n)}{n} = \prod_{(p,\_)\in P(n)} 1 - \frac{1}{p} \quad \geq \prod_{i=1}^{\log_2 n} \frac{i}{i+1} = \frac{1}{2} * \frac{2}{3} \cdots * \frac{\log_2 n}{1 + \log_2 n} = \frac{1}{1 + log_2 n}$$

where the final equality comes from pairing numerators and denominators in the expansion of the product, with only the first numerator and final denominator surviving.

We multiply by a constant, e.g., $\frac{1}{2}$, in the following inequality, and see that

$$\frac{1}{1 + \log_2 n} \geq \frac{1}{\log_2 n} * \frac{1}{2}$$

since $\log n \geq 1$ for all $n \geq 2$. Thus, $\frac{\phi(n)}{n} = \Omega\left(\frac{1}{\log n}\right)$, *Q.E.D.*

2.3 Given the prime factorization of $p-1$, it can be tested whether an $x \in Z_p^*$ is a generator in polynomial time in $\log p$

If $x \in Z_p^*$ is a generator, it must have order $p-1$. Assume x is **not** a generator. By Proposition 1 (and Lagrange's theorem), $\text{ord}(x) \mid p-1$, (but $p-1 \nmid \text{ord}(x)$). Thus $\text{ord}(x)$ shares some, but not all, prime factors with $p-1$. Assume p' is one prime factor of $p-1$ but not of $\text{ord}(x)$. Then $\text{ord}(x) \mid \frac{p-1}{p'}$, so $x^{(p-1)/p'} = 1$. To find out if x is a generator, we must therefore test this with all distinct primes in the prime factorization of $p-1$. If $\text{ord}(x) = p-1$ on the other hand, then $x^{(p-1)/k} \neq 1$ for any positive k, by the definition of order. So if we perform the above test with all distinct primes in the factorization of $p-1$ and find none of them gives $x^{(p-1)/p'} = 1$, then $\text{ord}(x) = p-1$, and x is a generator.

Dividing $p-1$ by p' can be done in polynomial time in $\log p$. To perform a single test, i.e., exponentiating x in Z_p^*, can be done in polynomial time in $\log p$ by binary exponentiation. Finally, the number of prime factors of $p-1$ can be at most $\log p$, since $p-1 = 2^{\log(p-1)} = p_1 p_2 ... p_\ell$, where the last expression is the prime factorization, with each $p_i \geq 2$.

Thus, testing whether x is a generator is done by performing a single test by dividing $p-1$ by a single prime factor and exponentiating in $O(P(\log p))$ time where $P(x)$ is a polynomial, and the test must be repeated up to $O(\log p)$ times, yielding a $O(\log p) * O(P(\log p))$ time bound, which is obviously polynomial in $\log p$, *Q.E.D.*.

3 An algorithm for finding generators of Z_p^*

Input p, a prime number and F, the facorization of $p-1$. Let $n = logp$ be the length of p, and thus the input size.

Output g, a generator of Z_p^*

Algorithm:

1. Randomly select an number g such that $1 \leq g \leq p-1$.
2. Test, using the approach in the proof of Proposition 3, whether g is a generator of Z_p^*.
3. If g is a generator, return g. Else, repeat from step 1.

The algorithm will only return correct values, since the final step only returns when we find exactly such an element that we are looking for, and not otherwise. The algorithm will probabilistically terminate, since there are always a generator of Z_p^* in Z_p^*.

Runtime analysis Randomly selecting an element can be done in constant time. Testing if it is a generator can be done in $\log p$ time by Proposition 3. There are $\phi(p-1)$ generators by Proposition 1, and $frac\phi(p-1)p-1 \geq \frac{1}{\log p-1}$ by Proposition 2, which means the so the chance of finding a generator at any given try is larger than $\frac{1}{\log p-1}$, because we are looking for $\phi(p-1)$ elements among $p-1$ elements. Thus, we expect to repeat a loop which takes $O(\log p)$ time for $\log p - 1 < \log p$ repetitions, so the algorithm's runtime $T = O(\log p) * O(\log p)$, which is $O(n^2)$.