Number Theory

EUnS

2019년 11월 28일

차 례

차 례 1
1 기초 정수론 2
2 유클리드 호제법(Euclidean algorithm) 3
3 확장된 유클리드 알고리즘(Extended Euclidean algorithm) 5
3.1 베주의 항등식 5
3.2 활용 5
4 나머지 연산에서 곱셈에 대한 역원 (modular multiplicative inverse) [1] 6
5 오일러의 ϕ함수(Euler's phi (totient) function) 7
6 오일러 정리(Euler's theorem)[2] 7
7 RSA시스템의 이해 9
7.1 RSA 공개 키 암호 시스템 [3] (RSA public-key cryptosystem) . . 9
7.2 공개키, 암호키 생성 9
7.3 단계 9
7.4 복호화 과정 9
7.5 이게 과연 안전한가? 10
8 중국인의 나머지 정리(Chinese Remainder Theorem) 11
9 소수 판별법 12
9.1 의사소수 판정(pseudoprime test) 12

[1] 역원: a와 연산자에 대해 연산결과가 항등원(= 1)이 되는 유일한 원소 b를 a의 역원이라한다.

[2] 페르마의 소정리는 오일러 정리에서의 특수한 경우이다.

[3] 로널드 라이베스트(Ron Rivest), 아디 샤미르(Adi Shamir), 레너드 애들먼(Leonard Adleman)이 세명의 이름 앞글자를 따서 지었다.

9.2 밀러라빈소수 판별법(Miller-Rabin primality test) 13
10 pollard's rho algorithms 14
11 advanced RSA 15
12 오일러의 ϕ함수(Euler's phi (totient) function) 15
13 오일러 정리(Euler's theorem)[4] 16

[4]페르마의 소정리는 오일러 정리에서의 특수한 경우이다.

1 기초 정수론

Theorem 1.1. $d, m, n,$이 어떤 정수일 때, 다음이 성립한다.

1. d가 m과 n의 공약수일때, $m+n$도 d의 배수이다.
2. d가 m과 n의 공약수일때, $m-n$도 d의 배수이다.

Proof. 이에 대한 증명은 간단합니다. $m = dq_1, n = dq_2 (q_1, q_2 \subset Z)$라 하자. $m+n = d(q_1+q_2), m-n = d(q_1-q_2)$

□

이 장을 이해하기위해서 약속 몇가지를 정의하겠습니다.

- d가 n의 약수(인수)일 때 $d \mid n$으로 표시합니다.
- m과 n의 최대공약수는 $\gcd(m, n)$이라고 합니다.
- r이 a를 b로 나눈 나머지라면 $r = a \bmod b$입니다.

이를써서 위 명제를 다시 적으면 $d \mid n, d \mid m \longrightarrow d \mid (m+n), d \mid (m-n)$

Theorem 1.2. a, b, z를 양의 정수라 하면, 다음이 성립한다.

$$ab \bmod z = [(a \bmod z)(b \bmod z)] \bmod z$$

Proof. $w = ab \bmod z$라 하자. 다음이 성립하는 q_1이 존재한다.

$$ab = q_1 z + w \Longleftrightarrow w = ab - q_1 z$$

마찬가지로 $x = a \bmod z, y = b \bmod z$라 하면, 다음을 만족시키는 q_2와 q_3, q가 존재한다.

$$a = q_2 z + x, b = q_3 z + y$$

$$\begin{aligned} w = ab - q_1 z &= (q_2 z + x)(q_3 z + y) - q_1 z \\ &= (q_2 q_3 z + q_2 y + q_3 x - q_1) z + xy \\ &= qz + xy \end{aligned}$$

여기서 $q = q_2q_3z + q_2y + q_3x - q_1$이므로

$$xy = -qz + w$$

즉 w는 xy를 z로 나눌 때의 나머지이다. 그러므로 $w = xy \bmod z$가 되고 이는 다음과 같이 나타낼 수 있다.

$$ab \bmod z = [(a \bmod z)(b \bmod z)] \bmod z$$

□

이는 큰수를 인수분해해서 작은값으로 나눠서 큰수를 다루는 부담을 덜어주지만 지수승에 대해서도 응용이 가능하다. 이를 이용해서 $a^{29} \bmod z$를 계산하는 절차를 예시로 들어보겠다. a^{29}는 다음과 같은 순서로 계산한다.

$$a, a^5 = a \cdot a^4, a^{13} = a^5 \cdot a^8, a^{29} = a^{13} \cdot a^{16}$$

$a^{29} \bmod z$는 다음과 같은 순서로 계산한다.

$$a \bmod z, a^5 \bmod z, a^{13} \bmod z, a^{29} \bmod z$$

$$\begin{aligned}
a^2 \bmod z &= [(a \bmod z)(a \bmod z)] \bmod z \\
a^4 \bmod z &= [(a^2 \bmod z)(a^2 \bmod z)] \bmod z \\
a^8 \bmod z &= [(a^4 \bmod z)(a^4 \bmod z)] \bmod z \\
a^{16} \bmod z &= [(a^8 \bmod z)(a^8 \bmod z)] \bmod z \\
a^5 \bmod z &= [(a \bmod z)(a^4 \bmod z)] \bmod z \\
a^{13} \bmod z &= [(a^5 \bmod z)(a^8 \bmod z)] \bmod z \\
a^{29} \bmod z &= [(a^{13} \bmod z)(a^{16} \bmod z)] \bmod z
\end{aligned}$$

2 유클리드 호제법(Euclidean algorithm)

Theorem 2.1. a가 음이 아닌 정수이고, b가 양의 정수이며 $r = a \bmod b$ 이면 다음이 성립한다.

$$\gcd(a, b) = \gcd(b, r)$$

수식이 익숙하지않은 분을 위해 풀어서 설명하자면, a가 음이 아닌 정수이고, b가 양의 정수이며, r이 a를 b로 나눈 나머지라면 a와 b의 최대공약수는 b와 r의 최대공약수와 같다.

Proof. $a = bq + r(0 \leq r < b, q$ 는 어떤 정수)인데, c를 a와 b의 공약수라 하면, c는 bq의 약수인 것은 자명하다. a또한 c의 약수이므로 c는 $a - bq\ (= r)$의 약수이다. 따라서 c는 b와 r의 공약수입니다. 반대로 c'가 b와 r의 공약수이면, c'는 $bq + r(= a)$의 약수가 되고 따라서 a와 b의 공약수가 된다다. 따라서 a와 b의 공약수 집합이 b와 r의 공약수 집합과 같으므로 $\gcd(a, b) = \gcd(b, r)$이 성립한다. □

유클리드 알고리즘의 의의는 나머지 연산만을 이용해서 뺑뺑이 돌리면 어떻게 됐든지 간에 최대공약수를 기계적으로 구할수있다는 것에 있다. $\gcd(a, b) = \gcd(b, r)$에서 b,r을 새로운 a,b로서 값을 넣어서 연속적으로 계산을 하면 언젠가 b가 0이 되는 순간이 오는데, 이때 a가 처음 a,b의 최대 공약수가 되는것이다.

Theorem 2.2. $\alpha > \beta$일때, 다음이 성립한다.
$\gcd(\alpha, \beta) = \gcd(\alpha - \beta, \beta)$

Proof. α, β의 최대 공약수를 x라 하자. $\alpha = x \cdot a, \beta = x \cdot b$ (a, b는 $a >$b이며 서로소인 두 정수)이며, $\alpha - \beta = x(a - b)$이다 $a - b$는 b와 서로소이며 두 값의 최대공약수는 여전히 x이다. □

Corollary 2.2.1. $f(n) = 1 + 10 + \cdots + 10^n$이라 하자.
$\gcd(f(x), f(y)) = f(\gcd(x, y))$임을 보여라.

Proof. $x > y$라 하자.

$$\begin{aligned} f(x) - f(y) &= 10^x + 10^{x-1} + \cdots + 10^{y+1} \\ &= 10^y(10^{x-y} + \cdots + 1) \\ &= f(x - y) \cdot 10^y \\ \gcd(f(x), f(y)) &= \gcd(f(x) - f(y), f(y)) \\ &= \gcd(f(x - y) \cdot 10^y, f(y)) \end{aligned}$$

이때 10^y와 $f(y)$는 항상 서로소이므로 $\gcd(f(x - y), f(y))$가 성립한다.

따라서 유클리드 호제법을 전개했을때, $\gcd(f(x), f(y)) = \gcd(f(\gcd(x, y)), 0)$ 이 되고 이는$f(\gcd(x, y))$과 같다. □

3 확장된 유클리드 알고리즘(Extended Euclidean algorithm)

확장된 유클리드 알고리즘은 다음의 방정식에대해서 s와 t를 효율적으로 구하는 방법에대한 것이다.

> a와 b가 음이 아니고 동시에 0이 아닌 정수라 하면 다음을 만족시키는 정수 s와 t가 존재한다.
>
> $$\gcd(a, b) = s \cdot a + t \cdot b^{a}$$
>
> [a]선형 디오판토스 방정식이라고도 한다.

3.1 베주의 항등식

Theorem 3.1. $ax + by = \gcd(x, y)$인 a, b가 존재한다.

Proof. 집합 $S = \{m | m = ax + by >, x \in \mathbf{Z}, y \in\}$를 생각해보면 ,이 집합 S는 $S \subset \mathbf{Z}$, $S \subset \varnothing$ (x, y를 원소로 가짐을 알 수 있다.) 이다. 또한, 자연수의 정렬성으로부터 최소가 되는 원소 d가 존재한다.

$\alpha \in S \Rightarrow \alpha = qd + r(0 \leq r < d)$라 하자.

만약 $d \nmid \alpha$ 일때, $r > 0$, $r = \alpha - qd, (\alpha, d \in S)$ $\alpha = a_1x + b_1y, d = a_2x + b_2y$ 라 하면. $r = (a_1 - a_2q)x + (b_1 - qb_2)y \in S$ $0 < r < d$인 r에 대해 d가 최소라는 가정이 모순이다.

$\therefore r = 0, d \mid \alpha(\forall \alpha \in S)$, $d \mid x, d \mid y \cdots$ d는 x , y의 공약수, $\gcd(x, y) = k$라 할때, $d = akx'' + bky'' = k(ax'' + by'')$ $k \mid d$에서 $k = d$ □

3.2 활용

이미 증명되어있는 유클리드 알고리즘의 흐름을 통해서 예시로 이해 해보자.
$a = 273$, $b = 110$으로 하는 $\gcd(273, 110)$을 구해봅시다.

$$r = 273 \bmod 110 = 53 \cdot \cdots 1$$

$a = 110, b = 53$으로 지정

$$r = 110 \mod 53 = 4 \cdots\cdots 2$$

$a = 53, b = 4$로 지정

$$r = 53 \mod 4 = 1 \cdots\cdots 3$$

$a = 4, b = 1$로 지정

$$r = 4 \bmod 1 = 0 \cdots\cdots 4$$

$r = 0$이므로 $\gcd(273, 110)$은 최대공약수로 1을 가진다. 여기서 *4* 식으로 되돌아가면 이는 다음과 같이 쓸 수 있다.

$$1 = 53 - 4 \cdot 13$$

계속 역순으로 뒤집어 올라가자 *3*

$$4 = 110 - 53 \cdot 2$$

이를 처음의 식에 대입하면

$$1 = 53 - (110 - 53 \cdot 2) \cdot 13 = 27 \cdot 53 - 13 \cdot 110$$

2

$$53 = 273 - 110 \cdot 2$$

이 식을 다시 대입하면

$$1 = 27 \cdot 53 - 13 \cdot 110 = 27 \cdot (273 - 110 \cdot 2) - 13 \cdot 110 = 27 \cdot 273 - 67 \cdot 110$$

따라서 $s = 27, t = -67$로서 성립하는 값을 찾았다.

4 나머지 연산에서 곱셈에 대한 역원 (modular multiplicative inverse) [5]

정의 4.1 (Inverse) $\gcd(n, \phi) = 1$인 두 정수 $n > 0, \phi > 1$가 있다고 하자.[a] $n \cdot s \bmod \phi = 1$을 만족시키는 s를 $n \bmod \phi$의 역원(inverse) 이라고 한다.

[a] 한 마디로 n과 ϕ는 서로소이다.

[5] 역원: a와 연산자에 대해 연산결과가 항등원(= 1)이 되는 유일한 원소 b를 a의 역원이라한다.

$\gcd(n, \phi) = 1$임을 이용해, 확장된 유클리드 알고리즘을 이용하여 $s' \cdot n + t \cdot \phi = 1$이되는 s'과 t'을 구할수있다. $n \cdot s' = -t'\phi + 1$이 되고 $\phi > 1$이므로 1이 나머지가 된다. $n \cdot s' \bmod \phi = 1$에서 $s = s' \bmod \phi$라 하면 $0 \leq s < \phi$가 되며 또한 $s \neq 0$이다.
위 식을 변형하면, $s' = q \cdot \phi + s$ 가되며 이를 만족하는 정수 q가 존재한다.
따라서

$$n \cdot s = ns' - \phi nq = -t'\phi + 1 - \phi nq = \phi(-t' - nq) + 1$$

따라서 $n \cdot s \bmod \phi = 1$이 된다.

5 오일러의 ϕ함수(Euler's phi (totient) function)

정의 5.1 (phi function) 양의 정수 n에 대해서
$\phi(n)$: 1부터 n까지의 양의 정수 중에 n과 서로소인 것의 개수를 나타내는 함수.

$\phi(n)$은 다음의 성질이 있다.

Theorem 5.1.

- 소수 p에 대해서 $\phi(p) = p - 1$
- m, n이 서로소인 정수일 때, 다음이 성립한다.

$$\phi(mn) = \phi(m)\phi(n)$$

성질이 몇 개 더 있지만 RSA에서 필요한것만을 다루기위해서 생략하였다.

Proof. 첫번째 성질은 어찌보면 당연하다 p는 소수이니 자기 자신을 제외한 모든 수와 서로소이다 (여기서 1도 세야한다.)
두번째 성질은 두수의 곱 mn은 각각 m에대해서 나눠지는 수가 n개이고 n에 대해서 나눠지는 수가 m개 이며 mn으로 나눠지는 수가 한 개이므로 $mn - \frac{mn}{m} - \frac{mn}{n} + \frac{mn}{mn} = mn - m - n + 1 = (m-1)(n-1) = \phi(m)\phi(n)$가 된다. □

6 오일러 정리(Euler's theorem)[6]

[6] 페르마의 소정리는 오일러 정리에서의 특수한 경우이다.

Theorem 6.1. 임의의 정수 a와 n이 서로소일 때, 다음이 성립한다.

$$a^{\phi(n)} \bmod n = 1$$

Proof. 정수 n에 대해서 1부터 n까지의 양의 정수 중에 n과 서로소인 것의 집합을 생각해보자. 그러면 이는 집합

$$A = \{r_1, r_2, r_3, \cdots, r_{\phi(n)}\}^7$$

으로 나타낼 수 있다. 이 집합은 A라하고 이 각 원소에 n과 서로소인 a를 곱한 집합을 B집합이라 하자.

$$B = \{ar_1, ar_2, ar_3, \cdots, ar_{\phi(n)}\}$$

확실한건 B에 있는 모든 원소는 n과 서로소인 것이다. 그럼 B집합의 각 원소를 mod$n$에 대해 계산한 것을 생각해보자. 이는 각 원소의 나머지가 a를 곱하기전 값과 같은지는 모르지만 $\phi(n)$개에 대해서 각각 일대일대응이 가능 한다는것을 알수있다. [8] 따라서 A의 모든 원소를 곱한 값에 $\bmod n$을 한것과 B의 모든 원소를 곱한 값에 mod$n$을 한 값은 같다.

$$ar_1 \cdot ar_2 \cdot ar_3 \cdots ar_{\phi_n} \equiv r_1 \cdot r_2 \cdot r_3 \cdots r_{\phi_n} \pmod{n}$$

$$a^{\phi(n)} \bmod n = 1$$

□

Corollary 6.1.1. 페르마의 소정리 소수 p에 대해 다음이 성립한다.

$$a^{p-1} \equiv 1 \pmod{p}$$

Proof. $\phi(p) = p - 1$이므로 오일러 정리에 도출된다. □

[7] 이러한 집합을 기약잉여계라고 부른다. 또한 집합 A의 원소의 갯수는 $\phi(n)$dl다.

[8] 실제 증명은 귀류법을 통해서 증명할수있다.$ar_i \equiv ar_j \bmod n$ 인 $1 \leq i < j \leq \phi(n)$ 이 존재한다고 가정해보자.

7 RSA시스템의 이해

7.1 RSA 공개 키 암호 시스템 [9] (RSA public-key cryptosystem)

이 알고리즘은 보안 기법중 하나로 가장 흔한 예시로서는 공인인증서가 있다.

$$A \longrightarrow B$$

A가 B에게 숫자를 하나 보낸다고 생각 해보자. A에게는 공개키가 필요하며 B에게는 개인키가 있어야한다. 공개키는 누가 가져도 상관없는 키이며 개인키는 절대로 노출되어서는 안되는 키이다.
A는 B에게 a를 보낼때 공개키를 이용하여 a를 c로 암호화 하여 보내며 B는 c를 공개키와 개인키를 이용하여 a로 복호화하여 읽는 방식이다.

7.2 공개키, 암호키 생성

두 개의 소수 p, q를 선택하여 $n = pq$를 계산한다. [10] 그 후 $\phi = (p-1)(q-1)$을 계산하고 $\gcd(n, \phi) = 1$인 정수 e을 선택한다. 그후 n과 e를 공개한다. $ed \bmod \phi = 1$이고 $0 < d < \phi$를 만족시키는 d를 생성하여 d를 개인키로 사용한다. [11]

7.3 단계

A가 B에게 정수 $a(0 \leq a \leq z-1)$를 보내기 위해서 A는 $c = a^e \bmod n$ 를 계산하여 c를 보낸다. [12] B는 $c^d \bmod n$를 계산하면 이 값이 a이다.

7.4 복호화 과정

$\phi(n) = \phi$

$ed \bmod \phi = 1 \Longleftrightarrow ed = b\phi(n) + 1$($b$는 어떤 상수)

[9] 로널드 라이베스트(Ron Rivest), 아디 샤미르(Adi Shamir), 레너드 애들먼(Leonard Adleman)이 세명의 이름 앞글자를 따서 지었다.

[10] 그 후 p, q는 버린다. 가지고 있어봤자 개인키가 뚫리는 취약점이 될수가있다.

[11] d는 위에서 언급한 나머지 연산에서 곱셈에 대한 역원을 구하는 방법으로 효율적으로 구할수 있다.

[12] c를 효율적으로 구하는 방법 또한 위에서 다루었다.

$$
\begin{aligned}
c^d \bmod n &= (a^e \bmod n)^d \bmod n \\
&= (a^e)^d \bmod n = a^{ed} \bmod n \\
&= a^{b\phi(n)+1} \bmod n \\
&= (a^{\phi(n)} \bmod n)^b a \bmod n = a
\end{aligned}
$$

[13]

7.5 이게 과연 안전한가?

이를 구하기위한 해결방법은 결과적으로 소인수분해와 직결되는데 그냥 n을 p와 q로소인수 분해해버리면 끝난다. 그러나 소인수분해를 다항시간내에하는 알고리즘은 개발되지 않았다.

[13] 오일러정리 사용

8 중국인의 나머지 정리(Chinese Remainder Theorem)

$x \equiv a_1 \pmod{m_1}$, $x \equiv a_2 \pmod{m_2}$, $\cdots$,
$x \equiv a_n \pmod{m_n} (\forall i, j \gcd(m_i, m_j) = 1^a)$ 일때, x가 $Z_{m_1 m_2 \cdots m_n}$에서 유일하게 존재한다.

[a] 서로소 아이디얼 (pairwise coprime)

Proof. 1. 존재성

$m = m_1 m_2 \cdots m_n, n_k = \dfrac{m}{n_k}$로 놓자. 그러면 $t_k m_k + s_k n_k = 1$인 정수 s_k, t_k가 존재한다 ($\because \gcd(m_k, n_k) = 1$) [14] $s_k n_k \equiv 1 \pmod{m_k}$ $x = a_1 n_1 s_1 + \cdot + a_n n_n s_n = \sum_{k=1}^{n} a_k n_k s_k$ $j \neq k \longrightarrow m_k \mid n_j \longrightarrow x \equiv a_k n_k s_k \equiv a_k \pmod{m_k}$

2. 유일성

귀류법을 사용한다. 서로다른 x, y가 $\bmod m$에서 합동식의 해라 하자. $x \equiv y \equiv a_1 \pmod{m_1}$ $x \equiv y \equiv a_2 \pmod{m_2}$ $x \equiv y \equiv a_n \pmod{m_n}$ $x - y \equiv 0 \pmod{m_k} (1 \leq k \leq n$인 정수 $k)$ $\mathrm{lcm}(m_1, m_2, \cdots, m_n) \mid (x - y) \longrightarrow m_1 m_2 \cdots m_n (= m) \mid (x - y) (\forall i, j \gcd(m_i, m_j) = 1) \therefore x \equiv y \pmod{\mathrm{lcm}(m_1, m_2, \cdots, m_n)}$ 이는 모순이다.

□

[14] 베주 항등식

9 소수 판별법

대표적인 방법

1부터 $\sqrt{n}$ 자연수까지 모두 나눠본후 나눠지는 수가없을시에 그수는 소수입니다. 2일때는 소수라 하고 짝수로 나누는 경우는 없애도 상관없다. $O(\sqrt{n})$

에라토스테네스의 체는 특정 n이 소수인지 판단하는것과는 무관하므로 제외합니다.

이글의 연장선상인 얘기입니다.

하나의 값 n을 놓고 이값이 소수인지 아닌지 보려면 현재 $O(\sqrt{n})$까지 만큼 비교해 보아야 확정적으로 알수있습니다.

엄청난 크기의 소수를 구하기 위해서 $O(\sqrt{n})$만큼의 시간도 길다고 판단해 이보다 좀더 효율적임을 위해서 결국 정확도를 조금 포기하고 확률적으로 소수인지 제대로 판단할 확률이 높은 알고리즘들이 나왔습니다.

9.1 의사소수 판정(pseudoprime test)

n이 소수일때 성립하는 페르마의 소정리를 판별방식으로 씁니다.

n과 서로소인 a에 대해서 $a^(n-1)$ 을 n으로 나눈 나머지는 무조건 1이 된다. 소수일때는 무조건 성립하니까 이를 판별 방식으로 쓰자는거죠

따라서 어떤 값 n에 대해서 $a^(n-1)$을 n으로 나눈 나머지가 1인지 판별해보면 됩니다. 적당한 a를 뽑고, $a^(n-1)$을 고속 지수승 알고리즘을 통해 $\log n$번에 구할수있다.

```
1 PSEUDOPRIME(n)
2     if MODULAR-EXPONENTIATION(2,n-1,n) != 1
3         return COMPOSITE
4     else return PRIME
```

이때 나오는 오진은 소수가 아닌데 $a^(n-1)$을 n으로 나눈 나머지가 1이 되는 경우이다. 이때 이 값을 **카마이클 수(Carmichael number)**라고 합니다 이 수의 특성도 재밌긴한데(사실 잘모름) 대충 넘어갑니다.

카마이클수를 차례대로 나타내면

561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, 41041, 46657, 52633, 62745, 63973

굉장히 띄엄띄엄 있습니다 따라서 오진율이 낮습니다.

9.2 밀러라빈소수 판별법(Miller-Rabin primality test)

앞의 의사소수 판정을 조금더 보완하기 위해서 검사를 더 촘촘히 하기로 합니다.

다음 증명된 사실을 가지고 합니다.

Theorem 9.1. 홀수인 소수 p와 정수 $1 \leq e$에 대해서 $\pmod{p^e}$에서 x^2을 n으로 나눈 나머지가 *1*이 되는 x의 해는 무조건 $1,-1(=n-1)$이다.

Proof. $p^e \mid (x+1)(x-1)$이 될때 p^e는 $(x-1)$,$(x+1)$둘 중에 하나만이 될수가 있다. p^e가 $(x+1),(x-1)$둘다 나눌수있을때에는 p^e가 2로 나누어지기 때문이다. 만약 p^e가 $x+1$을 나눌수있는 경우 $x \equiv -1 \pmod{p^e}$, p^e가 $x-1$을 나눌수있는 경우 $x \equiv 1 \pmod{p^e}$ 가된다. □

이 정리의 대우에 따라서 1,-1을 근으로 가지지 않은 n은 합성수라고 판단 할 수 있습니다.

따라서 이 나머지가 1이되는데 x가 +-1인지를 살펴 보면됩니다.

임의의 a에 대해서 판단하는 방법은 다음과 같습니다.

1. $n-1$ 를 2^{t*u} (d는 홀수)로 나타냅니다.
2. a^u부터 a^{2^t*u}로 점점 제곱하면서 이 사이에 값이 1,n-1인데 그전의 값이 +-1이 아닌지 판별을 합니다.
3. 마지막에 a^{2^t*u} 값이 1인지 비교합니다.(페르마 소정리)

예를 들어 카마이클 수는 561을 a = 2로 해서 구해보면 마지막에 $a^(2^t * d)$이 1이 되지만 제곱하기전의 값이 1이 아니라서 여기서 걸러지게 됩니다.

근데 이 판별이 결국 a에 따라 갈리게 됩니다. a값이 n에 대해서 제곱했을때 1이 되는 근이 아니어야 판별이 가능하죠. 이는 합성수 n에 대한 다음 판별 방식으로 탐지되는 근이 $1 \sim n-1$사이에 최소 n/2가 존재합니다

확실한건 a를 $2 \sim n/2$로 정하면 확실하게 나옵니다.

a를 촘촘하게 여러번 선택해서 판별하면 되는데 이러면 또 시간이 오래걸리죠

따라서 탐지율이 a를 뽑는 횟수에 따라 다릅니다.

알고리즘 복잡도는 a를 반복해서 뽑는 k에 따라서 $O(k \log^3 n)$이다. 추가적으로 곱셈을 FFT로 처리했을때 $O(k \log^2 n)$까지 줄일수있다.

추가로 난제중 하나인 리만가설이 맞다면 $2\log^2 n$ 개의 a로 검사를 했을때 소수일것이라고 판단이 되었을 경우 확정적으로 소수임이 드러나기때문에 $O(\log^4 n)$의 시간복잡도를 가지는 소수판별법이된다.

10 pollard's rho algorithms

의사난수를 발생시켜 구하는 방법이다....

$g(x) = x^2 - 1 \pmod{n}$

다음의 식을 사용해 반복적으로 $g(x)$를 만들어내고 $y = g(g(x))$로 $\gcd(y - x, n)$이 $1, n$인지 검사하는 방식이다. 그뒤 $x = g(x)$로 업데이트하여 반복한다.

11 advanced RSA

12 오일러의 ϕ함수(Euler's phi (totient) function)

정의 12.1 (phi function) 양의 정수 n에 대해서
$\phi(n)$: 1부터 n까지의 양의 정수 중에 n과 서로소인 것의 개수를 나타내는 함수.

$\phi(n)$은 다음의 성질이 있다.

Theorem 12.1.
- 소수 p에 대해서 $\phi(p) = p - 1$
- m, n이 서로소인 양의 정수일 때, 다음이 성립한다.

$$\phi(mn) = \phi(m)\phi(n)$$

- 소수 p와 양의 정수 α에 대해 다음이 성립한다.

$$\phi(p^\alpha) = p^\alpha \left(1 - \frac{1}{p}\right)$$

Proof. 첫번째 성질은 어찌보면 당연하다 p는 소수이니 자기 자신을 제외한 모든 수와 서로소이다 (여기서 1도 세야한다.)

두번째 성질은 두수의 곱 mn은 각각 m에대해서 나눠지는 수가 n개이고 n에 대해서 나눠지는 수가 m개 이며 mn으로 나눠지는 수가 한 개이므로 $mn - \frac{mn}{m} - \frac{mn}{n} + \frac{mn}{mn} = mn - m - n + 1 = (m-1)(n-1) = \phi(m)\phi(n)$가 된다.

세번째 성질 p^α보다 같거나 작은 p의 배수가 되는것 은 다음이 있다. $p, 2p, 3p, ..., p^{\alpha-1}p$ 따라서 총 $p^{\alpha-1}$개가 있고 $\phi(p^\alpha) = p^\alpha - p^{\alpha-1} = p^\alpha \left(1 - \frac{1}{p}\right)$

□

Corollary 12.1.1. *If $m_1, m_2, \ldots, m_k$ are k positive integers which are prime each to each, then*

$$\phi(m_1 m_2 \ldots m_k) = \phi(m_1)\phi(m_2)\ldots\phi(m_k).$$

If $m = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_n^{\alpha_n}$ where $p_1, p_2, \dots, p_n$ are different primes and $\alpha_1, \alpha_2, \dots, \alpha_n$ are positive integers, then

$$\phi(m) = m\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\dots\left(1 - \frac{1}{p_n}\right).$$

For,

$$\begin{aligned}\phi(m) &= \phi(p_1^{\alpha_1})\phi(p_2^{\alpha_2})\dots\phi(p_n^{\alpha_n}) \\ &= p_1^{\alpha_1}\left(1 - \frac{1}{p_1}\right) p_2^{\alpha_2}\left(1 - \frac{1}{p_2}\right)\dots p_n^{\alpha_n}\left(1 - \frac{1}{p_n}\right) \\ &= m\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\dots\left(1 - \frac{1}{p_n}\right).\end{aligned}$$

13 오일러 정리(Euler's theorem)[15]

Theorem 13.1. 임의의 정수 a와 n이 서로소일 때, 다음이 성립한다.

$$a^{\phi(n)} \bmod n = 1$$

Proof. 정수 n에 대해서 1부터 n까지의 양의 정수 중에 n과 서로소인 것의 집합을 생각해보자. 그러면 이는 집합

$$A = \{r_1, r_2, r_3, \cdots, r_{\phi(n)}\}^{16}$$

으로 나타낼 수 있다. 이 집합은 A라하고 이 각 원소에 n과 서로소인 a를 곱한 집합을 B집합이라 하자.

$$B = \{ar_1, ar_2, ar_3, \cdots, ar_{\phi(n)}\}$$

확실한건 B에 있는 모든 원소는 n과 서로소인 것이다. 그럼 B집합의 각 원소를 mod$n$에 대해 계산한 것을 생각해보자. 이는 각 원소의 나머지가 a를 곱하기전 값과 같은지는 모르지만 $\phi(n)$개에 대해서 각각 일대일대응이 가능 한다는것을 알수있다. [17] 따라서 A의 모든 원소를 곱한 값에 mod$n$을 한것과 B의 모든 원

[15] 페르마의 소정리는 오일러 정리에서의 특수한 경우이다.

[16] 이러한 집합을 기약잉여계라고 부른다. 또한 집합 A의 원소의 갯수는 $\phi(n)$dl다.

[17] 실제 증명은 귀류법을 통해서 증명할수있다.$ar_i \equiv ar_j \bmod n$ 인 $1 \le i < j \le \phi(n)$ 이 존재한다고 가정해보자.

소를 곱한 값에 mod$n$을 한 값은 같다.

$$ar_1 \cdot ar_2 \cdot ar_3 \cdots ar_{\phi_n} \equiv r_1 \cdot r_2 \cdot r_3 \cdots r_{\phi_n} \pmod{n}$$

$$a^{\phi(n)} \bmod n = 1$$

□

Corollary 13.1.1. 페르마의 소정리 *:* 소수 p에 대해 다음이 성립한다.

$$a^{p-1} \equiv 1 \pmod{p}$$

Proof. $\phi(p) = p - 1$이므로 오일러 정리에의해 성립함을 알 수 있다. □

$$a^{p-1} \equiv 1 \bmod p.$$

Then let us write

$$a^{p-1} = 1 + hp. \tag{1}$$

Raising each member of this equation to the p^{th} power we may write the result in the form

$$a^{p(p-1)} = 1 + h_1 p^2. \tag{2}$$

where h_1 is an integer. Hence

$$a^{p(p-1)} \equiv 1 \bmod p^2.$$

By raising each member of (2) to the p^{th} power we can readily show that

$$a^{p^2(p-1)} \equiv 1 \bmod p^3.$$

It is now easy to see that we shall have in general

$$a^{p^{\alpha-1}(p-1)} \equiv 1 \bmod p^{\alpha}.$$

where α is a positive integer; that is,

$$a^{\phi(p^\alpha)} \equiv 1 \bmod p^\alpha.$$

For the special case when p is 2 this result can be extended. For this case (1) becomes

$$a = 1 + 2h.$$

Squaring we have

$$a^2 = 1 + 4h(h+1).$$

Hence,

$$a^2 = 1 + 8h_1, \tag{3}$$

where h_1 is an integer. Therefore

$$a^2 \equiv 1 \bmod 2^3.$$

Squaring (3) we have

$$a^{2^2} = 1 + 2^4 h_2;$$

or

$$a^{2^2} \equiv 1 \bmod 2^4.$$

It is now easy to see that we shall have in general

$$a^{2^{\alpha-2}} \equiv 1 \bmod 2^\alpha$$

if $\alpha > 2$. That is,

$$a^{\frac{1}{2}\phi(2^\alpha)} \equiv 1 \bmod 2^\alpha \text{ if } a > 2. \tag{1}$$

Now in terms of the ϕ-function let us define a new function $\lambda(m)$ as follows:

$$\lambda(2^\alpha) = \phi(2^\alpha) \text{ if } a = 0, 1, 2;$$
$$\lambda(2^\alpha) = \frac{1}{2}\phi(2^\alpha) \text{ if } a > 2;$$
$$\lambda(p^\alpha) = \phi(p^\alpha) \text{ if } p \text{ is an odd prime;}$$
$$\lambda(2^\alpha p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_n^{\alpha_n}) = \text{lcm}(\lambda(2^\alpha), \lambda(p_1^{\alpha_1}), \lambda(p_2^{\alpha_2}), \ldots, \lambda(p_n^{\alpha_n}))$$

$2, p_1, p_2, \ldots, p_n$ being different primes.

Denote by m the number

$$m = 2^\alpha p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_n^{\alpha_n}.$$

Let a be any number prime to m. From our preceding results we have

$$a^{\lambda(2^\alpha)} \equiv 1 \bmod 2^\alpha,$$
$$a^{\lambda(p_1^{\alpha_1})} \equiv 1 \bmod p_1^{\alpha_1},$$
$$a^{\lambda(p_2^{\alpha_2})} \equiv 1 \bmod p_2^{\alpha_2},$$
$$\cdots$$
$$a^{\lambda(p_n^{\alpha_n})} \equiv 1 \bmod p_2^{\alpha_n}.$$

Now any one of these congruences remains true if both of its members are raised to the same positive integral power, whatever that power may be. Then let us raise both members of the first congruence to the power $\dfrac{\lambda(m)}{\lambda(2^\alpha)}$; both members of the second congruence to the power $\dfrac{\lambda(m)}{\lambda(p_1^{\alpha_1})}$; ...; both members of the last congruence to the power $\dfrac{\lambda(m)}{\lambda(p_n^{\alpha_n})}$. Then we have

$$a^{\lambda(m)} \equiv 1 \mod 2^\alpha,$$
$$a^{\lambda(m)} \equiv 1 \mod p_1^{\alpha_1},$$
$$\cdots\cdots$$
$$a^{\lambda(m)} \equiv 1 \mod p_n^{\alpha_n}.$$

From these congruences we have immediately

$$a^{\lambda(m)} \equiv 1 \mod m.$$

We may state this result in full in the following theorem:

If a and m are any two relatively prime positive integers, the congruence

$$a^{\lambda(m)} \equiv 1 \mod m.$$

is satisfied.

As an excellent example to show the possible difference between the exponent $\lambda(m)$ in this theorem and the exponent $\phi(m)$ in Fermat's general theorem, let us take

$$m = 2^6 \cdot 3^3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 19 \cdot 37 \cdot 73.$$

Here

$$\lambda(m) = 2^4 \cdot 3^2, \quad \phi(m) = 2^{31} \cdot 3^{10}.$$

In a later chapter we shall show that there is no exponent ν less than $\lambda(m)$ for which the congruence

$$a^{\nu} = 1 \mod m$$

is verified for every integer a prime to m.

From our theorem, as stated above, Fermat's general theorem follows as a corollary, since $\lambda(m)$ is obviously a factor of $\phi(m)$,

$$\phi(m) = \phi(2^{\alpha})\phi(p_1^{\alpha_1}) \dots \phi(p_n^{\alpha_n}).$$

31.8-2(CLRS)

It is possible to strengthen Euler's theorem slightly to the form
$a^{\lambda(n)} \equiv 1(\mod n)$ for all $a \in \mathbb{Z}_n^*$,
where $n = p_1^{e_1} \cdots p_r^{e_r}$ and $\lambda(n)$ is defined by

$$\lambda(n) = \text{lcm}(\phi(p_1^{e_1}), \dots, \phi\phi(p_r^{e_r})).$$

Prove that $\lambda(n) \mid \phi(n)$. A composite number n is a Carmichael number

if $\lambda(n) \mid n-1$. The smallest Carmichael number is $561 = 3 \cdot 11 \cdot 17$; here, $\lambda(n) = \text{lcm}(2, 10, 16) = 80$, which divides 560. Prove that Carmichael numbers must be both "square-free" (not divisible by the square of any prime) and the product of at least three primes. (For this reason, they are not very common.)

1. Prove that $\lambda(n) \mid \phi(n)$.

   $n = p_1^{e_1} \cdots p_r^{e_r}$

   $\phi(n) = \phi(p_1^{e_1}) * \ldots * \phi(p_r^{e_r})$

   $\text{lcm}(\phi(p_1^{e_1}, \ldots, \phi(p_r^{e_r}))|(\phi(p_1^{e_1}) * \ldots * \phi(p_r^{e_r}))$

   $\lambda(n) \mid \phi(n)$

2. Prove that Carmichael numbers must be both "square-free" (not divisible by the square of any prime)

   let Carmichael number $n = p^\alpha m(\alpha \geq 2, p \nmid m)$ $a^{n-1} \equiv 1 \pmod{n}(\gcd(a, n) = 1)$

   set $a = p + 1$ then $(p+1)^n \equiv p + 1 \pmod{n}$

   and $gcd(p^2, a) = 1$

   $(p+1)^n \equiv (p+1)^{p^2 p^{\alpha-2}} \equiv p + 1 \pmod{p^2}$

   but $\gcd(p^2, a) = 1$, $a \equiv 1 \pmod{p^2}$

   $p + 1 \equiv 1 \pmod{p^2}$ This is impossible

   https://math.stackexchange.com/questions/1764812/carmichael-number-square-free

3. the product of at least three primes.

   Assume that $n = pq$, with $p < q$ two distinct primes, is a Carmichael number. Then we have $qa1 \pmod{q`1)} \rightarrow n \equiv pq \equiv p \pmod{q`1} \rightarrow n`1 \equiv p`1 \pmod{q`1}$ Here $0 < p`1 < q`1$ , so $n`1$ is not divisible by $q`1$.

   https://math.stackexchange.com/questions/432162/carmichael-proof-of-at-least-3-factors

2. Prove that Carmichael numbers must be both "square-free" (not divisible by the square of any prime)

https://math.stackexchange.com/questions/1764812/carmichael-number-square-free

Proof. 카마이클 수 $n = p^{\alpha}m(\alpha \geq 2, p \nmid m)$라 하자 정의에 따라 다음이 성립한다. $a^n \equiv a \pmod{n}$.

$a \equiv 1 + p \pmod{p^{\alpha}}$ 라 하자.

$(p+1)^n \equiv p+1 \pmod{m}$이 되는데. m의 인자인 p^2와에 대해서도 다음이 성립한다. $(p+1)^n \equiv (p+1)^{p^{\alpha}m} \equiv (p+1)^{p^2 p^{\alpha-2}m} \equiv p+1 \pmod{p^2}$ 그러나 $\gcd(a,n) = 1$ 이라서 $\gcd(a,p^2) = 1$이다. $(p+1)^n \equiv p+1 \equiv 1 \pmod{p^2}$이며 이는 모순이다. □

3. the product of at least three primes.

https://math.stackexchange.com/questions/432162/carmichael-proof-of-at-least-3-factors

Assume that $n = pq$, with $p < q$ two distinct primes, is a Carmichael number. Then we have $qa1 \pmod{q`1}) \rightarrow n \equiv pq \equiv p \pmod{q`1} \rightarrow n`1 \equiv p`1 \pmod{q`1}$ Here $0 < p`1 < q`1$ , so $n`1$ is not divisible by $q`1$.

http://www.gutenberg.org/files/13693/13693-pdf.pdf

Carmichael function

$\lambda(n)$ 다음을 만족하는 가장 작은 양의 정수 m

$a^m \equiv 1 \pmod{n}$

$n = p_1^{e_1} \cdots p_r^{e_r}$

$\lambda(n) = \mathrm{lcm}(\phi(p_1^{e_1}), \ldots, \phi(p_r^{e_r}))$