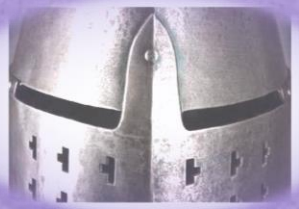


Chapter 2

Introduction to Number Theory

Divisibility

- All operations are about integers
- Division: a÷b
  - a: dividend（被除數）
  - b: divisor (除數）
- b divides a (b 整除 a)
  - if a = mb for some integer m (quotient 商數)
  - Notation: b | a
- If b|a, b is a divisor of a

Properties of Divisibility

- If $a \mid 1$, then $a = \pm 1$
- If $a \mid b$ and $b \mid a$, then $a = \pm b$
- b|0, for an $b \neq 0$
- If $a \mid b$ and $b \mid c$, then $a \mid c$
  - 11|66 and 66|198 → 11|198
- If $b \mid g$ and $b \mid h$, then $b \mid (mg + nh)$ for any integers m and n

Greatest Common Divisor

- gcd(*a*, *b*): the **greatest common divisor** of *a* and *b*
- gcd(*a*, *b*) = gcd(|*a*|, |*b*|)
- gcd(a, b) = gcd(b, a)
- gcd(a, 0) = |a|
- b | a → gcd(a, b) = b
- *a* and *b* are **relatively prime** if gcd(*a*, *b*) = *1*

Modular Arithmetic

- "a mod n": the remainder when *a* is divided by *n*.
  - a mod n = r = a – q*n*, where q = $\lfloor a/n \rfloor$ and $0 \leq r < n$
  - q: quotient
  - n >0: modulus
  - a = q*n* +*r*
- Examples
  - 11 mod 7 = 11 - $\lfloor 11/7 \rfloor$ x 7 = 4
  - -11 mod 7= -11 - $\lfloor -11/7 \rfloor$ x 7 = 3
- n | (a-r)

Modular Arithmetic: properties

- $(a + b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n$
- $(a - b) \bmod n = [(a \bmod n) - (b \bmod n)] \bmod n$
- $(a * b) \bmod n = [(a \bmod n) * (b \bmod n)] \bmod n$

Congruence

- a and b are **congruent modulo n**
  if $(a \bmod n) = (b \bmod n)$
- Notation: $a \equiv b \pmod{n}$
- Examples
  - $73 \equiv 27 \pmod{23}$
  - $21 \equiv -9 \pmod{10}$

Congruence properties

- $a \equiv b \pmod{n}$ ➔ $n \mid (a - b)$
- $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$
  ➔ $a \equiv c \pmod{n}$

Modular Multiplicative Inverse

- $x=a^{-1} \bmod n$ *if* $ax \bmod n=1$
- *Examples*
  - $3^{-1} \bmod 7 = 5$ $(3 \times 5) \bmod 7 = 1$
  - $4^{-1} \bmod 15 = 4$ $(4 \times 4) \bmod 15 = 1$
  - …

GCD principle

- Integers a and b, a > b > 0

Theorem: gcd(a, b) = gcd(b, a mod b)

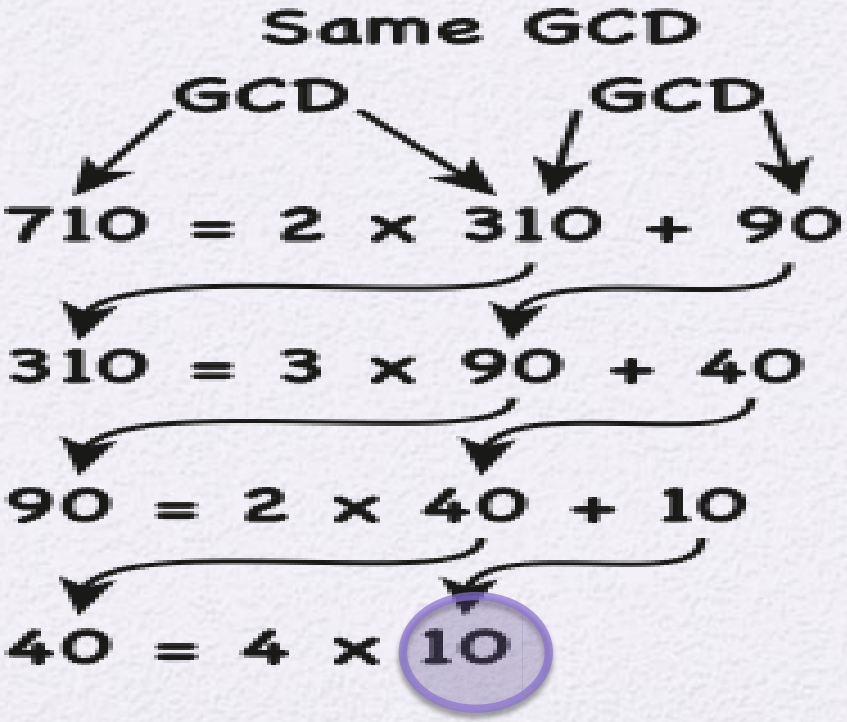
(輾転相除法)

Proof:

- Let c=gcd(a,b).
- c|a and c|b ➔ c|(a-bq). Thus, c|gcd(b, r). (a-bq = r)
- If c'=gcd(b, r) > c, let c'=mc, m>1.
- mc|b and mc|(a-bq) ➔ mc|a. This contradicts gcd(a,b)=c.
- Thus, c'=c=gcd(b, a mod b)

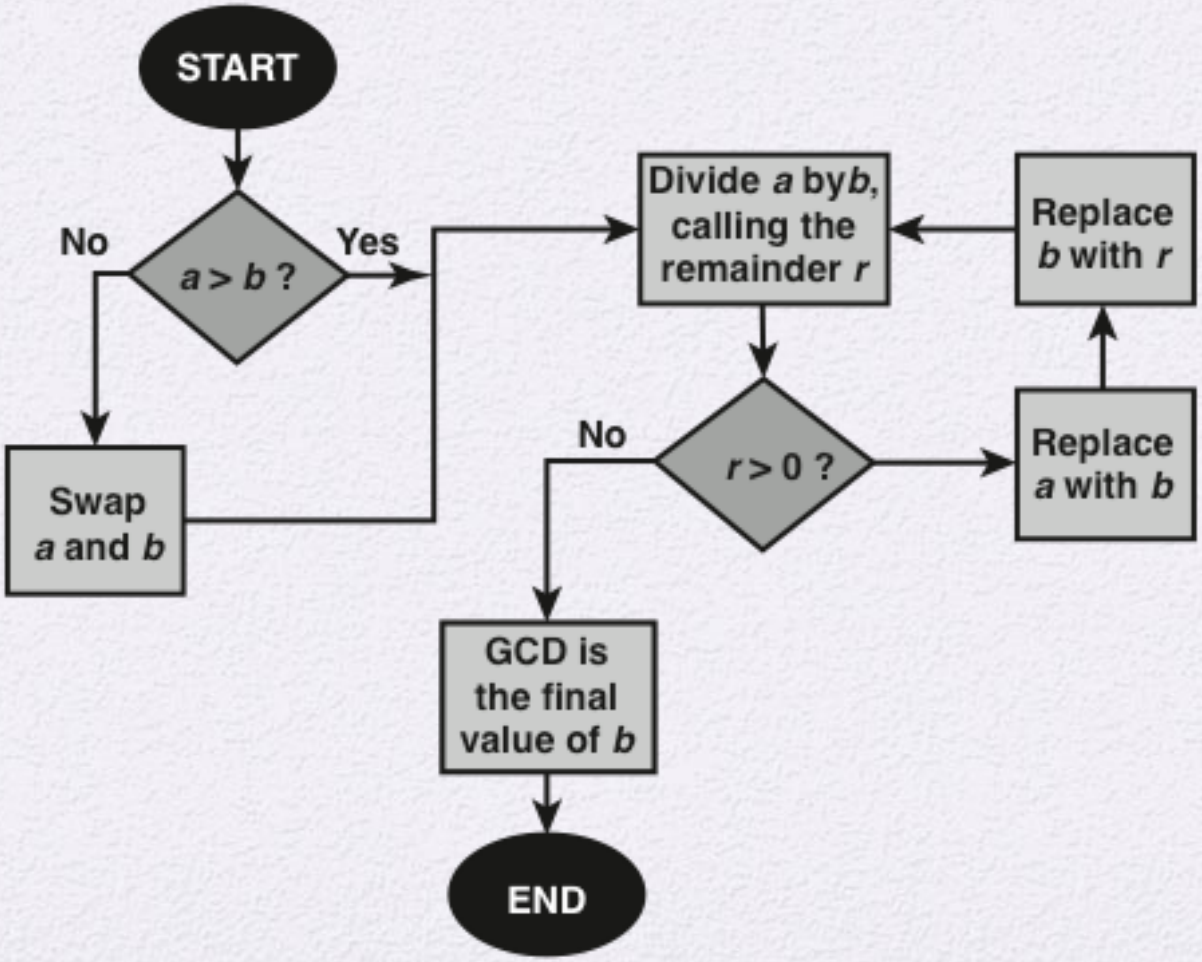
Euclidean algorithm

- Compute gcd(710, 310)



Euclidean algorithm

a, b >0



Euclidean Algorithm: Example

Dividend	Divisor	Quotient	Remainder
$a = 1160718174$	$b = 316258250$	$q_1 = 3$	$r_1 = 211943424$
$b = 316258250$	$r_1 = 211943424$	$q_2 = 1$	$r_2 = 104314826$
$r_1 = 211943424$	$r_2 = 104314826$	$q_3 = 2$	$r_3 = 3313772$
$r_2 = 104314826$	$r_3 = 3313772$	$q_4 = 31$	$r_4 = 1587894$
$r_3 = 3313772$	$r_4 = 1587894$	$q_5 = 2$	$r_5 = 137984$
$r_4 = 1587894$	$r_5 = 137984$	$q_6 = 11$	$r_6 = 70070$
$r_5 = 137984$	$r_6 = 70070$	$q_7 = 1$	$r_7 = 67914$
$r_6 = 70070$	$r_7 = 67914$	$q_8 = 1$	$r_8 = 2156$
$r_7 = 67914$	$r_8 = 2156$	$q_9 = 31$	$r_9 = 1078$
$r_8 = 2156$	$r_9 = 1078$	$q_{10} = 2$	$r_{10} = 0$

Fast !!!

Complexity

- a, b: n-bit long
- Worst case: $\log_2 b = n$ calls of gcd's
- Each gcd step needs to compute "x mod y",
- x mod y
  - Division by the shift-substract method
  - O(n)
- Total: $O(n^2) = O((\log_2 a)^2)$

Extended Euclidean Algorithm

- Given a and b, find integers x and y for

$$xa + yb = gcd(a, b)$$

ex. $21x + 12y = 3$

r	q	x	y
21	x	0	1
12	x	1	0
9	1	-1	1
3	1	2	-1

- Steps
  - Initial
    - 1a+0b=a → Eq. I
    - 0a+1b=b →Eq. II
  - Compute a mod b: a=qb+r
  - Compute (I) - q(II) = (1)a - (q)b = r → Eq. III
  - Consider: Eq. II and III for gcd(b, r)
  - Continue until r=0

⇒ $x = 2$, $y = -1$ #

An example

a=1759, b=550 $x_i a + y_i b = r_i$ for all i

i	r_i	q_i	x_i	Y_i
–1	1759		1	0
0	550		0	1
1	109	3	1	–3
2	5	5	–5	16
3	4	21	106	–339
4	1	1	–111	355
5	0	4		

gcd(a,b)

- –111 a + 355 b = 1

Find modular inverse

(当 gcd(a,b)=1 时可以找乘法反元素)

- Given a and n, gcd(a,n)=1, find

$$x = a^{-1} \bmod n$$

$$\begin{cases} ax+by=1 \\ x = a^{-1} \bmod n \end{cases}$$

- x is unique within [1..n-1]
- Method
  - Find integers x and y for xa + yn = 1 = gcd(a,n) by the extended Euclidean algorithm
  - x is a^{-1} mod n since xa mod n = 1-yn mod n = 1

Prime Numbers

- A prime number has only two divisors of 1 and itself
- Any integer $a > 1$ can be factored in a unique way as

$$a = p_1^{a1} \times p_2^{a2} \times \ldots \times p_t^{at}$$

where $p_1 < p_2 < \ldots < p_t$ are prime numbers and where each a_i is a positive integer

- This is known as the fundamental theorem of arithmetic

Primes Under 2000

2	101	211	307	401	503	601	701	809	907	1009	1103	1201	1301	1409	1511	1601	1709	1801	1901
3	103	223	311	409	509	607	709	811	911	1013	1109	1213	1303	1423	1523	1607	1721	1811	1907
5	107	227	313	419	521	613	719	821	919	1019	1117	1217	1307	1427	1531	1609	1723	1823	1913
7	109	229	317	421	523	617	727	823	929	1021	1123	1223	1319	1429	1543	1613	1733	1831	1931
11	113	233	331	431	541	619	733	827	937	1031	1129	1229	1321	1433	1549	1619	1741	1847	1933
13	127	239	337	433	547	631	739	829	941	1033	1151	1231	1327	1439	1553	1621	1747	1861	1949
17	131	241	347	439	557	641	743	839	947	1039	1153	1237	1361	1447	1559	1627	1753	1867	1951
19	137	251	349	443	563	643	751	853	953	1049	1163	1249	1367	1451	1567	1637	1759	1871	1973
23	139	257	353	449	569	647	757	857	967	1051	1171	1259	1373	1453	1571	1657	1777	1873	1979
29	149	263	359	457	571	653	761	859	971	1061	1181	1277	1381	1459	1579	1663	1783	1877	1987
31	151	269	367	461	577	659	769	863	977	1063	1187	1279	1399	1471	1583	1667	1787	1879	1993
37	157	271	373	463	587	661	773	877	983	1069	1193	1283		1481	1597	1669	1789	1889	1997
41	163	277	379	467	593	673	787	881	991	1087		1289		1483		1693			1999
43	167	281	383	479	599	677	797	883	997	1091		1291		1487		1697			
47	173	283	389	487		683		887		1093		1297		1489		1699			
53	179	293	397	491		691				1097				1493					
59	181			499										1499					
61	191																		
67	193																		
71	197																		
73	199																		
79																			
83																			
89																			
97																			

Fermat's Little Theorem

- Theorem: If p is prime and a is a positive integer not divisible by p, then

$$a^{p-1} \equiv 1 \pmod{p} \quad \Rightarrow a^{p-1} \bmod p = 1$$

- An alternate form is:
  - If p is prime and a is a positive integer, then

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

Euler's Totient Function

$$\phi(n) = p_1^{a1-1} p_2^{a2-1} \cdots p_t^{at-1} \times (p_1 - 1) \cdots (p_t - 1)$$

$$= |\{a : 0 < a < n, \gcd(a, n) = 1\}|$$

n	$\phi(n)$
1	0
2	1
3	2
4	2
5	4
6	2
7	6
8	4
9	6
10	4

n	$\phi(n)$
11	10
12	4
13	12
14	6
15	8
16	8
17	16
18	6
19	18
20	8

n	$\phi(n)$
21	12
22	10
23	22
24	8
25	20
26	12
27	18
28	12
29	28
30	8

Euler's Theorem

- An extension of Fermat's little theorem
- For every *a* and *n*, *gcd(a, n)=1*

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

- An alternative form is, for a, n: positive integers

$$a^{\phi(n)+1} \equiv a \pmod{n}$$

Primality Test: Theory

Given n , determine whether n is prime

- Instead of testing "n being prime" directly, we test "n being composite"
- If the test of "n being composite" fail for many times, then n is **probably** prime.

Primality Test: Theory

if $x^2 \bmod n = 1$

then check whether $x \bmod n = \pm 1$

until x cannot go sqrt or $n \neq 1$

- If n is prime,
  - F1: $x^{n-1} \bmod n = 1$ for any x, $0<x<n$
  - F2: $1 = x^2 \bmod n$ has only trivial solutions 1 and -1.
- Find a witness for "n is composite"
  - By F1, if we find x with $x^{n-1} \bmod n \neq 1$, then 'n is composite'
  - By F2, if we find a non-trivial solution x (non- ± 1) for Eq: $1 = x^2 \bmod n$, then 'n is composite'

Primality Test: Theory

ex. $n=17$

$F_1: 2^{16} \bmod 17 = 1$

$F_2: 2^8 \bmod 17 = 1 \quad (8 = 2^3)$

$F_3: 2^4 \bmod 17 = -1 \quad (4 = 2^2)$

How to find such a witness?

- Let n-1 $=2^r d$ even, where d is odd
- Randomly pick a, 1<a<n
- If $a^{n-1} \bmod n \neq 1$, we found a witness for F1.
- Else
  - If $a^{2^{i-1}d} \neq \pm 1$ and $a^{2^i d}=1$ for some i, then $a^{2^{i-1}d}$ is a non-trivial solution for $1=x^2 \bmod n$. (For F2)

Facts

- If n is not prime, for at least a half of a's, we can find a witness.
- We can try many random a's and cannot a witness, the probability of n being prime is high.

Primality test: Rabin-Miller

Input: n: odd

1. Let $n-1=2^r d$, d is odd
2. Pick a random number a, $1< a < n$
   - If $a^{n-1} \bmod n \neq 1$, return(n is composite)
   - Compute $b_0 = a^d \bmod n$, $b_1 = a^{2\times d} \bmod n, \ldots,$ $b_r = a^{2^r \times d} \bmod n$
   - If $b_{i-1} \neq \pm 1$ and $b_i = 1$, return(n is composite)
3. Try step 2. t times. If none return "n is composite", then return(n is probably prime)

Primality test: success probability

- Theorem: If n is composite,

  $|\{a \mid 0<a<n, \text{find a witness from this } a\}| \geq 3(n-1)/4$

- Thus,
  - Pr[RM(n) is probably prime | n is prime] = 1
  - Pr[RM(n) is probably prime | n is composite] $\leq (1/4)^t$
- Fast!!!

n=69

n-1=68=2^2x17

- Randomly pick a =47

1. a^{68} mod 69 = 1
2. a^{34} mod 69 = 1
3. a^{17} mod 69 = 47

the next is not ±1

→ 47 is a witness. ⇒ composite

Output (69 is composite)

n=37

n-1 = 36 = 2^2x9

- Random a=4

1. a^{36} mod 37 =1
2. a^{18} mod 37 = 1
3. a^9 mod 37 = 36 = -1

→ no witness is found

- Random a=2

1. a^{36} mod 37 =1
2. a^{18} mod 37 = 36 = -1
3. a^9 mod 37 = 311 (not important)

→ no witness is found

- Try more random 'a'

…

Since we found no witnesses for F1 and F2, output (37 is probably prime)

Deterministic Primality Algorithm

- Prior to 2002 there was no known method of efficiently proving the primality of very large numbers
- All of the algorithms in use produced a probabilistic result
- In 2002 Agrawal, Kayal, and Saxena developed an algorithm that efficiently determines whether a given large number is prime
  - Known as the AKS algorithm
  - Does not appear to be as efficient as the Miller-Rabin algorithm

Chinese Remainder Theorem (CRT)

中國餘式定理

- Believed to have been discovered by the Chinese mathematician Sun-Tsu in around 100 A.D.
- One of the most useful results of number theory
- Find x for the system of linear modulus equations:

$$x \bmod m_i = r_i, \quad 1 \le i \le k.$$

where $\gcd(m_i, m_j)=1$, for all $i \neq j$

ex. $\begin{cases} X \equiv 5 \pmod 7 \\ X \equiv 3 \pmod 9 \\ X \equiv 6 \pmod{16} \end{cases}$

CRT Solutions

- $M = m_1 m_2 \ldots m_k$
- $M_i = m_1 \ldots m_{i-1} m_{i+1} \ldots m_k = \frac{M}{m_i}$
- $C_i = M_i^{-1} \bmod m_i$
  - $M_i C_i \bmod m_i = 1$
  - $M_i \bmod m_j = 0$, for $j \neq i$

By this definition

$$\begin{cases} x \equiv r_1 \pmod{m_1} \\ x \equiv r_2 \pmod{m_2} \\ \vdots \\ x \equiv r_k \pmod{m_k} \end{cases}$$

- $x = r_1 M_1 C_1 + r_2 M_2 C_2 + \ldots + r_k M_k C_k + tM$, for any t
- There is a unique solution in [0 .. M-1]
- Fast!!!

CRT Solutions: example

$(56 \times 2) \% 3 = 1$

- Find solutions for
  - $2 = x \bmod 3$
  - $3 = x \bmod 7$
  - $2 = x \bmod 8$

$M_1 = 7 \times 8 = 56$ $\quad M_1^{-1} \bmod 3 = 2$

$M_2 = 3 \times 8 = 24$ $\quad M_2^{-1} \bmod 7 = 5$

$M_3 = 3 \times 7 = 21$ $\quad M_3^{-1} \bmod 8 = 5$

- $M_1 = 56$, $M_2 = 24$, $M_3 = 21$
- $M_1^{-1} \bmod 3 = 2$, $M_2^{-1} \bmod 7 = 5$, $M_3^{-1} \bmod 8 = 5$
- $x = (2 \times 56 \times 2 + 3 \times 24 \times 5 + 2 \times 21 \times 5) \bmod 168 = 122$

Modular exponentiation

- Given a, g and p, compute

$$y = g^a \bmod p$$

- By the square and multiply algorithm
- Example: compute $g^{11} \bmod p$
  - Compute $g^2 \bmod p$, $g^4 \bmod p$, $g^8 \bmod p$
  - Compute $g^{11} \bmod p = g x g^2 x g^8 \bmod p$

Powers of Integers, Modulo 19

g^1	g^2	g^3	g^4	g^5	g^6	g^7	g^8	g^9	g^{10}	g^{11}	g^{12}	g^{13}	g^{14}	g^{15}	g^{16}	g^{17}	g^{18}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
3	9	8	5	15	7	2	6	18	16	10	11	14	4	12	17	13	1
4	16	7	9	17	11	6	5	1	4	16	7	9	17	11	6	5	1
5	6	11	17	9	7	16	4	1	5	6	11	17	9	7	16	4	1
6	17	7	4	5	11	9	16	1	6	17	7	4	5	11	9	16	1
7	11	1	7	11	1	7	11	1	7	11	1	7	11	1	7	11	1
8	7	18	11	12	1	8	7	18	11	12	1	8	7	18	11	12	1
9	5	7	6	16	11	4	17	1	9	5	7	6	16	11	4	17	1
10	5	12	6	3	11	15	17	18	9	14	7	13	16	8	4	2	1
11	7	1	11	7	1	11	7	1	11	7	1	11	7	1	11	7	1
12	11	18	7	8	1	12	11	18	7	8	1	12	11	18	7	8	1
13	17	12	4	14	11	10	16	18	6	2	7	15	5	8	9	3	1
14	6	8	17	10	7	3	4	18	5	13	11	2	9	12	16	15	1
15	16	12	9	2	11	13	5	18	4	3	7	10	17	8	6	14	1
16	9	11	5	4	7	17	6	1	16	9	11	5	4	7	17	6	1
17	4	11	16	6	7	5	9	1	17	4	11	16	6	7	5	9	1
18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1

Discrete logarithm

- Find x for given y, g and prime p, 1 <g <p,

$$y = g^x \bmod p \quad \text{or} \quad x = \text{dlog}_{g,p}(y)$$

- Generalized version

底数 mod值

$$y = g^x \bmod n \quad \text{or} \quad x = \text{dlog}_{g,n}(y)$$

Tables of Discrete Logarithms, Modulo 19

(a) Discrete logarithms to the base 2, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{2,19}(a)$	18	1	13	2	16	14	6	3	8	17	12	15	5	7	11	4	10	9

(b) Discrete logarithms to the base 3, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{3,19}(a)$	18	7	1	14	4	8	6	3	2	11	12	15	17	13	5	10	16	9

(c) Discrete logarithms to the base 10, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{10,19}(a)$	18	17	5	16	2	4	12	15	10	1	6	3	13	11	7	14	8	9

(d) Discrete logarithms to the base 13, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{13,19}(a)$	18	11	17	4	14	10	12	15	16	7	6	3	1	5	13	8	2	9

(e) Discrete logarithms to the base 14, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\log_{14,19}(a)$	18	13	7	8	10	2	6	3	14	5	12	15	11	1	17	16	4	9

Discrete logarithm: complexity

- Best time for computing dlog is $e^{[(\ln p)^{1/3} (\ln\ln p)^{2/3}]}$
  - "ln p" is the length of prime p
  - We use p of thousands of bits
- Thus, computing $x=\mathrm{dlog}_{g,p}(a)$ is very difficult for very long p !!!
- However, computing $g^a \bmod p$ is fast by the "square-and-multiply" algorithm

Summary

- Divisibility and the division algorithm
- The Euclidean algorithm
  - Greatest Common Divisor
  - Finding the Greatest Common Divisor
- Modular arithmetic
  - The modulus
  - Properties of congruence
  - Modular arithmetic operations
  - Properties of modular arithmetic
  - Euclidean algorithm revisited
  - The extended Euclidean algorithm
- Prime numbers
  - Fermat's Theorem
  - Euler's totient function
  - Euler's Theorem
  - Testing for primality
  - Miller-Rabin algorithm
- Chinese Remainder Theorem
- Discrete logarithms
  - Powers of an integer, modulo n
  - Logarithms for modular arithmetic
  - Calculation of discrete logarithms