

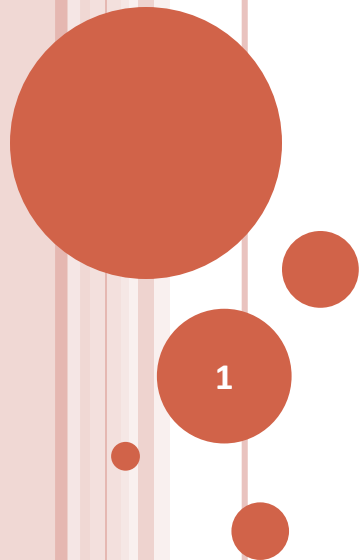
PROBLEM DOMAIN – NUMBER THEORY

Modular Arithmetic:

Groups Z_n, Z_n^*

Size of Z_n^*

Computing the size of Z_n^*



CONGRUENCE ARITHMETIC

- “congruence modulo n ”:
 - if $a \bmod n = b \bmod n$
 - then a and b are congruent modulo n
 - This is an equivalence relation.
 - Why?
 - This is often denoted as
 - $a \equiv b \pmod{n}$



CONGRUENCE ARITHMETIC - \mathbb{Z}_N

- $(\mathbb{Z}_n = \{ 0, 1, \dots, n-1 \}, +_n)$ is a group
 - where $+_n$ refers to addition modulo n .
- **Exercise:** *Verify the following properties:*
 - Closure:
 - Associativity:
 - Existence of Identity :
 - 0
 - Existence of Inverse:
 - a^{-1} is $n-a$



CONGRUENCE ARITHMETIC: \mathbb{Z}_N^*

○ $(\mathbb{Z}_n^* = \{x \mid 1 \leq x \leq n \text{ and } \gcd(x, n) = 1\}, *_n)$ is a group.

- **Exercise:**

- *Verify Closure and Associativity*

- Identity Element exists:

- $(a * 1 = a)$

- Inverse?



CONGRUENCE ARITHMETIC: \mathbb{Z}_n^*

[CONTD.]

- $(\mathbb{Z}_n^* = \{x \mid 1 \leq x \leq n \text{ and } \gcd(x, n) = 1\}, *_n)$ is a group.
- Existence of Inverse:
 - Is there an x such that $a * x = 1 \pmod{n}$, for a in \mathbb{Z}_n^* ?
 - i.e. Is there an x such that $a * x = 1 + b * n$ for some +ve integer b ?
 - The answer is yes, by extended Euclid's Theorem since $\gcd(a, n) = 1$
 - Furthermore, by Aryabhatia's algorithm:
 - *the inverse of any element in $(\mathbb{Z}_n^*, *_n)$ can be computed in polynomial time i.e.*
 - *time that is polynomial in $\log(n)$*



CONGRUENCE ARITHMETIC – SIZE OF Z_N^*

○ What is the size of Z_n^* ?

- Let $\phi(n)$, known as *Euler's phi function*, denote the size of Z_n^*

○ Properties of $\phi(n)$

- $\phi(p) = p-1$ for prime p

- Proof: for any $m < p$, $\gcd(m, p) = 1$ for prime p .

- $\phi(p^m) = p^m - p^{m-1}$ for prime p

- Proof:

- Only multiples of p have common factors with p^m

- Multiples of p (less than p^m) are:

- $p, 2*p, 3*p, \dots, (p^{m-1} - 1) * p$

- So, $\phi(p^m) = (p^m - 1) - (p^{m-1} - 1)$ for prime p .

CONGRUENCE ARITHMETIC - PROPERTIES OF $\phi(N)$ [CONTD]

- $\phi(p*q) = (p-1)(q-1)$ for primes p and q

Proof:

- Only multiples of p or q or both have common factors with $p*q$
 - i.e. $p, 2*p, \dots, q*p$, and $q, 2*q, \dots, p*q$
- And they are all distinct except for $p*q$
- So $\phi(p*q) = (p*q) - p - q + 1$
- ϕ is multiplicative i.e.
 - $\phi(m*n) = \phi(m) * \phi(n)$ if $\gcd(m,n) = 1$
 - Proof: *Left as an exercise.*
 - **Note:** We only need $\phi(p^{k1} * q^{k2}) = \phi(p^{k1}) * \phi(q^{k2})$ for primes p and q . **End of Note.**

CONGRUENCE ARITHMETIC – COMPUTING $\phi(N)$

○ Value of $\phi(n)$

- If $n = p_1^{k_1} * p_2^{k_2} * \dots * p_m^{k_m}$ for primes p_i and +ve integers k_i then $\phi(n) = \prod_i (p_i^{k_i} - p_i^{k_i-1})$

○ Computing $\phi(n)$

- If the prime factors of n are known then $\phi(n)$ can be computed in polynomial time
 - But computing factors is known to be “**difficult**”.
 - In particular, there is no known polynomial time algorithm to compute factors of a given integer.
- Is there an alternative?
 - i.e can we compute $\phi(n)$ – efficiently – without computing factors of n ?

CONGRUENCE ARITHMETIC – COMPUTING $\phi(n)$ – SPECIAL CASE

- Can we compute $\phi(n)$ – efficiently – without computing factors of n ?
- Consider $n = p * q$
 - Given n and $\phi(n)$, one can compute p and q in polynomial time !
 - How?
 - i.e. *Computing $\phi(n)$ is at least as “difficult” as factoring n .*



ASIDE: REDUCTION AND LOWER BOUNDING

- We argued that one problem (say, computing $\phi(n)$) is ***at least as difficult to solve as*** another problem (say, factoring n):
 - but we argued this without solving – i.e. without providing an algorithm for – either of these problems independently!!

- In the abstract, we argued that:

- given an algorithm f for problem π_1 ,
- if we can construct an algorithm $g \bullet f$ for problem π_2
- such that g costs no more than f

○ then we can conclude that π_1 is ***at least as difficult as*** π_2

This is referred to as ***lower-bounding*** (the cost / complexity of a problem)

This construction is referred to as ***reduction*** i.e. we reduce π_2 to π_1