Advanced Data Structures and Algorithms
Lecture 2: Number Theory(Cont.)

Mostafa Mahmoud

German University in Cairo

16 February, 2013

Outline

- Group Axioms
- Multiplicative groups, subgroups, Euler's totient function
- Powers of an element
- Lagrange, Euler, Fermat theorems
- Multiplicative inverse, Binary Exponentiation
- Primitive multiplicative roots
- Discrete Logarithm
- Baby-step Giant-step Algorithm

Group Axioms

A group G, is an ordered pair $(S, \odot)$ of a set S, and a binary function $\odot : S \times S \to S$, that satisfies 3 axioms. The binary function is called the operator of the group.

- Associative: $(a \odot b) \odot c = a \odot (b \odot c)$
- Identity: There exists a unique item $e \in S$, such that for all $a \in S$ we have that $a \odot e = e \odot a = a$.
- Inverse: For all $a \in S$, there exists an item $b \in S$, such that $a \odot b = b \odot a = e$. By the first 2 axioms, the inverse of each element is unique, thus the inverse of a will be denoted by a^{-1}.

The group is Abelian group, if the operator is commutative.

- Commutative: If $a, b \in S$, then $a \odot b = b \odot a$.

Multiplicative groups, subgroups, Euler's totient function

- A group $H = (R, \odot)$ is a subgroup of a group $G = (S, \odot)$, only if $R \subseteq S$ and the operator $\odot$ is closed under the set R, that is, if $a, b \in R$, then $a \odot b \in R$.
- If we have a natural number n, we consider the group $((\mathbb{Z}/n\mathbb{Z})^*, \odot_n)$ as the multiplicative group modulo n. It is defined by

$$(\mathbb{Z}/n\mathbb{Z})^* := \{a \in \{1, \cdots, n\} | \gcd(n, a) = 1\}$$

,

$$a \odot_n b = ab \mod n$$

- Euler's totient function $\phi(n)$, is defined to be the size of the multiplicative group modulo n.

$$\phi(n) = |(\mathbb{Z}/n\mathbb{Z})^*|$$

Euler's totient function

- ϕ is a multiplicative function, which implies that if n, m are coprimes, then $\phi(mn) = \phi(m)\phi(n)$ and $\phi(1) = 1$.
- If we have a prime p and a natural number $e > 0$, then $(\mathbb{Z}/p^e\mathbb{Z})^*$ will contain all numbers that are not multiples of p, so $\phi(p^e) = p^e - p^{e-1} = p^{e-1}(p-1)$
- Using the previous two statements, we can deduce that

$$\phi(p_1^{e_1} \cdots p_r^{e_r}) = p_1^{e_1-1} \cdots p_r^{e_r-1} \cdot (p_1 - 1) \cdots (p_r - 1)$$

Powers of an element

- If we have a group $G = (S, \odot)$ and $a \in S$, then we denote the set $\langle a \rangle$ as the set generated by the element a;

$$\langle a \rangle := \{a^0, a^1, \cdots\}$$

- Any group generated by an element a, is a subgroup of the multiplicative group modulo n.
- If we have an element $a \in S$, then $(a^{-1})^n = (a^n)^{-1} = a^{-n}$. This can be proved by mathematical induction.
- If the group generated by an element a is finite, then it's cyclic, we will then denote the order of group by $|\langle a \rangle|$, which is the number of distinct elements in the set $\langle a \rangle$.

Powers of an element

- The first repeated element in a group generated by an element a, will be a^m, given that $|\langle a \rangle| = m$.
- If $|\langle a \rangle| = m$, then $a^m = e$.
- Proof: (By Contradiction)
  - On the contrary, we will assume that $a^m = a^n$ for some $n > 0$, then we can multiply both sides by a^{-n}, arriving at $a^{m-n} = e$, which is a contradiction that the $|\langle a \rangle = m|$.
- If $|\langle a \rangle| = m$, then for all $k < m$, we have $a^{-k} = a^{m-k}$.

Lagrange's Theorem

- If we have a subgroup $H = (R, \odot)$ of a group $G = (S, \odot)$, then $|\langle H \rangle|$ divides $|\langle G \rangle|$.
- Theorem: Any element $a \in S$, will be contained in one of the cosets of H, $\{g \odot H | g \in S\}$
  - Since H is a subgroup, then it has the identity element e, then $a \in aH$, thus every element will be in at least one of the cosets.
- Theorem: Any two cosets of H are either the same or disjoint.
  - If two cosets aH, bH have an intersection, then there are elements $x, y \in H$ such that $ax = by \Rightarrow b^{-1}a = yx^{-1}$.
  - Since x, y are in a subgroup H, then $yx^{-1} \in H$ so as $b^{-1}a$.
  - If $g \in aH$, then $g = az$ for some $z \in H$; we have that $b^{-1}az \in H$, so we arrive at $g = b(b^{-1}az) \Rightarrow g \in bH$. We $aH \subseteq bH$, without loss of generality we get $bH \subseteq aH$.

Lagrange's Theorem

- Theorem: All cosets of H have the same size.
- Proof:
  - For two cosets aH, bH, the function $f : aH \to bH$ is bijective, $f(x) = ba^{-1}x, f^{-1}(y) = ab^{-1}y$, thus any two cosets have the same size.
- By using the previous theorems, we get that all the cosets of the subgroup H have the same size, then the size of any subgroup divides the size of the super group.

Euler's Theorem and Fermat's little theorem

- Euler's Theorem: Given two coprime natural number a, n, then
$$a^{\phi(n)} \equiv 1 \pmod{n}$$
  - Proof: Since a, n are coprimes, then $a \in (\mathbb{Z}/n\mathbb{Z})^*$, then by Lagrange's theorem $\phi(n) = k|\langle a \rangle|$ for some k; then
$$a^{\phi(n)} = (a^{|\langle a \rangle|})^k = 1^k = 1$$
- Fermat's little theorem: Given a prime number p and an integer a, then
$$a^p \equiv a \pmod{p}$$
  - Proof: if p doesn't divide a, then by Euler's theorem $a^{\phi(p)} = a^{p-1} = 1$, so $a^p \equiv a \pmod{p}$; else $a^p \equiv a \equiv 0 \pmod{p}$

Multiplicative inverse, Binary Exponentiation

- Given two coprime integers a, n, then a^{-1} is the multiplicative inverse of a, it can be computed in two ways.
  - Bezout's identity, using extended euclidean algorithm, we get $ax + ny = 1$, thus $ax \equiv 1 \pmod{n}$, then $a^{-1} = x$.
  - Using Euler's theorem, The element $a^{-1} = a^{\phi(n)-1}$, as by Euler's theorem $a \odot_n a^{\phi(n)-1} \equiv 1 \pmod{n}$
- One problem remains in method 2, computing a power of an element efficiently, we can do this by binary exponentiation.
  - Computing $a^n \mod m$ can be computed recursively by computing $(a^{\lfloor n/2 \rfloor} \mod m)^2 \cdot a^{n \mod 2} \mod m$

Primitive multiplicative roots

- An element $a \in (\mathbb{Z}/n\mathbb{Z})^*$ is a primitive root modulo n, if $\langle a \rangle = (\mathbb{Z}/n\mathbb{Z})^*$.
- Theorem: If a is a primitive root modulo n, then for all divisors $d > 1$ of $\phi(n)$ this formula holds

$$a^{\phi(n)/d} \not\equiv 1 \pmod{n}$$

- Proof: (By Contradiction)
  - On the contrary, we will assume there is a divisor $d > 1$, such that $a^{\phi(n)/d} \equiv 1 \pmod{n}$, this implies that the order of the subgroup generated by a is a divisor of $\phi(n)/d$, which implies that $|\langle a \rangle| < \phi(n)$, which contradicts that $\langle a \rangle = (\mathbb{Z}/n\mathbb{Z})^*$

Primitive multiplicative roots

- We will use this theorem to test if a given $a \in (\mathbb{Z}/n\mathbb{Z})^*$ is a primitive root modulo n.
- It is sufficient to check only for the prime divisors of $\phi(n)$, instead of all the divisors. For any divisor d, there is a prime divisor p of $\phi(n)$ that divides d, so if $a^{\phi(n)/d} \equiv 1 \pmod{n}$, then $a^{\phi(n)/p} \equiv 1 \pmod{n}$.
- The only numbers that have a primitive multiplicative root, are on the form $1, 2, 4, p^i, 2p^i$ for an odd prime p, and $i > 0$.

Discrete Logarithm

- Theorem: If g is a primitive root modulo n, then

$$g^x \equiv g^y \pmod{n} \iff x \equiv y \pmod{\phi(n)}$$

- Given an element $a \in (\mathbb{Z}/n\mathbb{Z})^*$ and g primitive root modulo n, we are faced with a problem for which power x does the equality $g^x = a$ hold.
- We will use the Baby-Step Giant-Step to find the value of x. This algorithm is a meet-in a middle algorithm, it is a modification of the trial multiplication algorithm. This algorithm has a time complexity $O(\sqrt{m})$, while the trial multiplication has time complexity $O(m)$, where $m = \phi(n)$.

Baby-step Giant-step Algorithm

- Given a group of size n, generated by an element g, and we have an element a in this group. We want to find a number x, such that $g^x = a$
- We will set $m := \lceil \sqrt{n} \rceil$ and $\gamma = g^{-m}$, then we can represent any number x such that $0 <= x < n$ as $x = im + j$ for some i, j such that $0 <= i, j < m$.
- Then we have $g^x = g^{im+j} = a$, then $g^j = \gamma^i \cdot a$.
- The algorithm inserts all the values $g^0, g^1, \cdots, g^{m-1}$ into a hash map, and then we loop on the values $a \cdot \gamma^0, a \cdot \gamma^1, \cdots, a \cdot \gamma^{m-1}$, and check if any of them is in the hash map.

Sources

- https://en.wikipedia.org/wiki/Group_(mathematics)#Definition_and_illustration
- https://en.wikipedia.org/wiki/Primitive_root_modulo_n
- https://en.wikipedia.org/wiki/Lagrange%27s_theorem_(group_theory)
- https://en.wikipedia.org/wiki/Euler%27s_totient_function
- https://www.topcoder.com/community/data-science/data-science-tutorials/primality-testing-non-deterministic-algorithms/
- Abstract Algbera https://www.extension.harvard.edu/open-learning-initiative/abstract-algebra