Introduction to advanced number theory

- Many results in computer science, especially in cryptography, are reliant on number theory. Prime numbers, modular arithmetic, and relatively prime pairs are fundamental concepts.
- Computer science and cryptographic applications often involve computations in modular arithmetic, specifically solving equations. The module aims to explore the capabilities and limitations of modular arithmetic.
- The Euclidean algorithm is a method for computing the GCD of two numbers. It involves a series of remainders until reaching a remainder of 0. The last non-zero remainder is the GCD.
- The core module presented a theorem stating that if $a \equiv c \pmod p$ and $b \equiv d \pmod p$ then certain operations like addition and multiplication also follow the modular congruence.
- This module also raises the question of how to perform algebraic operations, particularly division, in modular arithmetic. It encourages exploring rules and understanding 'division' in this context.
- The module aims to help students understand division in the context of modular arithmetic, discover algebraic rules within this framework, and address challenges associated with performing algebraic operations in modular arithmetic.
- This number theory is useful in Cryptographic application, algorithm efficiency, computational mathematics, problem solving in computer science and many more.

Bézout's lemma and the extended Euclidean algorithm

- Bézout's lemma states that for any integers a and b, if gcd(a, b) = c, there exist numbers m and n such that $m \cdot a + n \cdot b = c$.
- If gcd(a, b) = 1, then there exist m and n such that $m \cdot a + n \cdot b = 1$, implying a and b are relatively prime.
- Bézout's lemma is closely tied to the Euclidean algorithm, which is used to find the greatest common divisor (gcd) of two numbers.
- The video also shows the application of Bézout's lemma in finding Bézout coefficients for various pairs, demonstrating how to express the gcd as a linear combination of the numbers.

- The extended Euclidean algorithm is an extension of the Euclidean algorithm and serves as its algorithmic counterpart. It's utilized to find Bézout's coefficients (s and t).
- In the extended algorithm, two extra parameters, s and t, are introduced. Initially, s corresponds to a, and t corresponds to b.
- The algorithm involves a series of substitutions and rearrangements, maintaining s and t values, until the remainder becomes 0. The coefficients obtained from the process express the gcd as a linear combination of a and b.
- Demonstrated through examples, the extended Euclidean algorithm is applied to find the gcd and Bézout coefficients for different pairs of numbers.
- It is used in modular arithmetic to divide and to reduce fractions to their most basic form. This algorithm's computations are a component of the cryptographic protocols that safeguard the internet. it is also used in Understanding the GCD Computation and Solving Algebraic Problems

Modular inverse

- In algebra, dividing by x is equivalent to multiplying by 1/x. The modular inverse of a number x mod p, denoted as y, satisfies $xy \equiv 1 \pmod{p}$.
- A number m has an inverse mod p if and only if m and p are relatively prime ($\gcd(m, p) = 1$).
- The proof involves Bézout's lemma. If m has an inverse mod p (denoted as n), then $mn \equiv 1 \bmod p$, and m and n are relatively prime. Conversely, if m is relatively prime with p, Bézout's lemma ensures the existence of the inverse.
- To find the inverse of a number m mod p, apply the extended Euclidean algorithm, find the Bézout coefficient for m, and the result is the modular inverse.
- Demonstrated examples include finding the inverse of 3 mod 17 using the extended Euclidean algorithm and applying the concept to solve modular inverse problems for various numbers.
- If $\gcd(m, p) = 1$, the equation $mx \equiv a \pmod{p}$ has a solution for any number x. The solution is obtained by multiplying both sides by the modular inverse of m.
- For a prime number p, any linear equation $ax \equiv a \pmod{p}$ has a solution, except when $a \equiv 0 \bmod p$. The proof relies on the fact that if $a \not\equiv 0 \bmod p$, then $\gcd(a, p) = 1$.
- A video further explains the process of solving linear equations in modular arithmetic and provides a solution to a specific exercise.
- This is also useful in Cryptographic application, algorithm efficiency, computational mathematics, problem solving in computer science and many more.

Fermat's and Euler's theorems

- Fermat's Little Theorem states that for a prime number p and a number a not divisible by p, $a^p \equiv a \pmod{p}$.
- The proof involves observing that the remainders of $k \cdot a$ for k from 1 to $p-1$ are all distinct and cover the range 1 to $p-1$.
- The multiplication of both sides simplifies to $1 \pmod{p}$, leading to the conclusion of Fermat's Little Theorem.
- Euler's Theorem is an extension of Fermat's Theorem for cases when p is not prime.
- It involves Euler's totient function $\phi(p)$, representing the count of numbers between 1 and $p-1$ that are relatively prime to p.
- Euler's Totient Theorem states that for any 'a' such that $\gcd(a,p)=1$, $a^{\phi(p)} \equiv 1 \pmod{p}$.
- The proof is analogous to Fermat's, with the set S being $\{m \in \{1 \ldots p-1\} : \gcd(m,p)=1\}$.
- Both theorems can be used to quickly compute powers of a modulo p.
- They can also be applied to compute the inverse of a modulo p.
- For calculating powers, Euler's Theorem allows us to reduce the exponent modulo $\phi(p)$, making computations more efficient.
- Some usefulness of this are: Application in Cryptography, Reducing Exponentiation, Inverse Calculation, Efficient Power Computation.

References

1. *LinkedIn Login, Sign in | LinkedIn*. (n.d.). LinkedIn. https://www.linkedin.com/pulse/number-theory-everyday-life-geetha-muthu/
2. Wikipedia contributors. (2023, December 28). *Euclidean algorithm*. Wikipedia. https://en.wikipedia.org/wiki/Euclidean_algorithm#:~:text=It%20is%20used%20for%20reducing,by%20factoring%20large%20composite%20numbers.