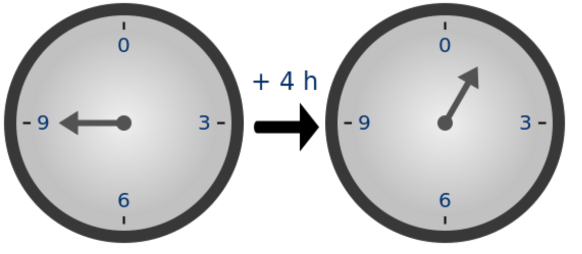
Math
Problem Solving Club Nov 9 2016

N mathematicians walk into a bar. The first orders 1 beer, the second orders ½ a beer, the third orders ¼ a beer and so on

The bartender says stupid mathematicians and gives them 2 beers

Modular arithmetic

- Modular arithmetic is a system of arithmetic for integers, where numbers "wrap around" upon reaching a certain value—the modulus
- If $a \equiv b \pmod{m}$, a and b are said to be **congruent** modulo m
- $1 \equiv 13 \pmod{12}$
- $1 \equiv 25 \pmod{12}$
- $-1 \equiv 11 \pmod{12}$

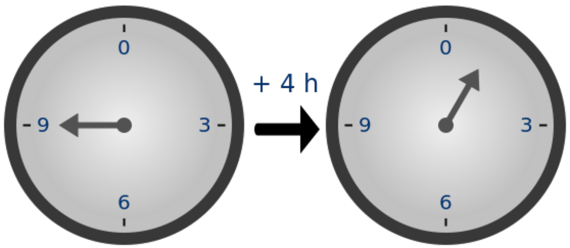


Properties of modular arithmetic

- Which of the following is true?

  If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$

- $a+c \equiv b+d \pmod{m}$ ✓
- $a-c \equiv b-d \pmod{m}$ ✓
- $a \times c \equiv b \times d \pmod{m}$ ✓
- $a^c \equiv b^d \pmod{m}$ ✗
- $a^c \equiv a^d \pmod{m}$ ✗
- $a^c \equiv b^c \pmod{m}$ ✓
- $a \div c \equiv b \div d \pmod{m}$ ✗



Modulo operation

- In computing, the modulo operation finds the remainder after division of one number by another
- 25 % 12 =
  - Answer: 1
- (-1) % 12 =
  - In C++/Java: -1
  - In Python: 11
- Problems frequently want the answer **modulo m**, which usually means the non-negative remainder when the answer is divided by m.
- (a+b)%m = (a%m + b%m)%m
- (a-b)%m = (a%m - b%m)%m
- (a*b)%m = ((a%m)*(b%m))%m

Greatest common divisor

- gcd(a, b) is the largest integer that divides both a and b
- For example, gcd(8, 12) = 4
- What is gcd(60, 45)?
- How do you compute gcd(a, b)?
  - **Euclidean algorithm**
  - function gcd(a, b)
  -     if b = 0
  -         return a;
  -     else
  -         return gcd(b, a % b);

Exponentiation by squaring

- How can we calculate a^b?
- Naive exponentiation: O(b)
- Observe that x^n =
  - If n is even, then $(x^{n/2})^2$
  - If n is odd, then $x(x^{(n-1)/2})^2$
- What is the time complexity of evaluating?
  - O(log b)
- This can be also used for raising matrices to high powers (e.g. finding the n’th Fibonacci number)

Modular inverse

- $(a/b) \% m \neq ((a\%m) / (b\%m)) \% m$
- How can we do modular division?
  - We can **sometimes** use a modular inverse
- If a^{-1} is the modular multiplicative inverse of a modulo m, then $aa^{-1} = 1 \pmod{m}$
  - Now $(a/b) \% m = ((a\%m) * ((b^{-1})\%m)) \% m$
- When does the modular inverse exist?
  - The multiplicative inverse of a modulo m exists if and only if a and m are coprime (i.e., if gcd(a, m) = 1).

Finding the modular inverse

- How do we compute modular inverses?
- Approach 1: Extended euclidean algorithm
  - Generally the fastest and easiest approach
  - A slightly modified version of the Euclidean algorithm can find modular inverses
- Approach 2: Euler’s (or Fermat’s little) theorem
  - $a^{\varphi(m)-1} \equiv a^{-1} \pmod{m}$ where $\varphi(m)$ is Euler's totient function (positive integers up to a given integer n that are relatively prime to n)
  - For a prime modulus p, $a^{p-2} \equiv a^{-1} \pmod{p}$
  - Use exponentiation by squaring

Logarithms

- Useful properties of logarithms:
  - $\log(a \times b) = \log a + \log b$
  - $\log(a \div b) = \log a - \log b$
  - $\log(a^b) = b \log a$
- How do you find the number of digits in a number?
  - $\log_{10}(1) = 0$
  - $\log_{10}(2) \approx 0.3010$
  - $\log_{10}(999) \approx 2.9996$
  - $\log_{10}(1000) = 3$
  - $\log_{10}(1001) \approx 3.0004$
- The number of digits in n is $\lfloor \log_{10}(n) \rfloor + 1$