Problem Solving (Number Theory)

Srivaths P

Binary Exponentiation

The idea of binary exponentiation is as follows:

When B is even: $A^B = A^{\frac{B}{2}} \times A^{\frac{B}{2}}$.
When B is odd: $A^B = A^{\frac{B}{2}} \times A^{\frac{B}{2}} \times A$.
(Assuming division is floored)

We can do the above using a recursive function (or iteratively).

Greatest Common Divisor

$GCD(A,\ B)$ is the Greatest Common Divisor of A and B.

$LCM(A,\ B)$ is the Least Common Multiple of A and B.

To calculate GCD efficiently, we can use the Euclidean Algorithm.

Euclidean Algorithm states that $GCD(A,\ B) = GCD(B\ \%\ A,\ A)$.

When $A = 0$, the solution is B.

Euclidean Algorithm – Code

Recursive:

```
int gcd_(int a, int b) {
    if (a == 0) return b;
    return gcd_(b%a, a);
}
```

Iterative:

```
int gcd_(int a, int b) {
    while (a) {
        int t = a;
        a = b % a;
        b = t;
    }

    return b;
}
```

Least Common Multiple

$LCM(A, B)$ can be calculated as

$$\frac{A \times B}{GCD(A, B)}$$

(Only works for 2 numbers)

Properties of GCD / LCM

- $GCD(A, B)$ can be represented as product of $\min(p_i^{a_i}, p_i^{b_i})$ for each prime factor.
- $LCM(A, B)$ can be represented as product of $\max(p_i^{a_i}, p_i^{b_i})$ for each prime factor.
- $GCD(A, B, C, \ldots)$ is the same as $GCD(GCD(GCD(A,\ B),\ C),\ \ldots)$
- $LCM(A, B, C, \ldots)$ is the same as $LCM(LCM(LCM(A,\ B),\ C),\ \ldots)$
- $GCD(A,\ B) \times LCM(A,\ B) = A \times B$
- $GCD(A,\ A+1) = 1$

OEIS - Online Encyclopedia of Integer Sequences

Link: https://oeis.org/

OEIS can be used to find the formula of an integer sequence with just the first few values (which could be computed using brute-force or manually by hand).

Problem Solving

- https://cses.fi/problemset/task/1081
- https://codeforces.com/problemset/problem/1474/B
- https://codeforces.com/problemset/problem/1471/A
- https://codeforces.com/problemset/problem/1617/B

Thanks for Watching!

Feedback form:
https://forms.gle/LiCo5Ckj252M7NFU6