# ChatGPT - Number Theory for FAANG

Give detailed notes on Number Theory for FAANG preparation. Include all patterns/formulas with clear explanations from basic to advanced.

### Comprehensive Notes on Number Theory for FAANG Preparation

Number theory is a critical topic in competitive programming and interviews. It involves the study of integers and their properties, often focusing on prime numbers, divisors, modular arithmetic, and patterns.

---

## 1. Basics of Number Theory

### 1.1 Divisors

- **Definition:** If $a$ divides $b$ (denoted as $a|b$), then there exists an integer $k$ such that $b = a \times k$.
- **Properties:**
  - $1$ and the number itself are always divisors of a number.
  - If $a|b$ and $b|c$, then $a|c$.
- **Finding Divisors:** Iterate up to $\sqrt{n}$ to find all divisors. For every $i$ such that $i|n$, both $i$ and $n/i$ are divisors.

**Example:**

```text
Divisors of 36: 1, 2, 3, 4, 6, 9, 12, 18, 36
```

---

### 1.2 Prime Numbers

- **Definition:** A prime number is greater than 1 and has only two divisors: 1 and itself.

- **Smallest prime number:** 2 (also the only even prime).
- **Properties:**
  - If $n$ is not prime, it has a divisor less than or equal to $\sqrt{n}$.
  - The total number of primes less than $n$ can be approximated using the **Prime Number Theorem**: $\pi(n) \approx \frac{n}{\ln(n)}$.

---

### 1.3 Euclidean Algorithm (GCD)

- **Greatest Common Divisor (GCD):** The largest number that divides both $a$ and $b$.
- **Algorithm:**

$$\mathrm{GCD}(a, b) = \mathrm{GCD}(b, a\%b)$$

  Repeat until $b = 0$, then $\mathrm{GCD}(a, 0) = a$.

**Code:**

```java
int gcd(int a, int b) {
    return b == 0 ? a : gcd(b, a % b);
}
```

---

### 1.4 Least Common Multiple (LCM)

- **Formula:**

$$\mathrm{LCM}(a, b) = \frac{a \times b}{\mathrm{GCD}(a, b)}$$

---

## 2. Modular Arithmetic

## 2.1 Basic Operations

- **Definition:** Modular arithmetic deals with remainders.

$$a \bmod m = r \quad \text{where } r = a - m \times \text{floor}(a/m)$$

- **Properties:**

1. $(a + b) \mod m = [(a \mod m) + (b \mod m)] \mod m$
2. $(a - b) \mod m = [(a \mod m) - (b \mod m)] \mod m$
3. $(a \times b) \mod m = [(a \mod m) \times (b \mod m)] \mod m$
4. **Modular Exponentiation:** Efficiently computes $a^b \mod m$ using repeated squaring.

---

## 2.2 Modular Inverse

- **Definition:** $x$ is the modular inverse of $a \mod m$ if:

$$(a \times x) \mod m = 1$$

- **Finding Modular Inverse:**

1. Using Extended Euclidean Algorithm.
2. Using Fermat's Little Theorem if $m$ is prime:

$$a^{m-1} \equiv 1 \pmod m \implies a^{m-2} \equiv a^{-1} \pmod m$$

---

# 3. Prime Numbers in Depth

## 3.1 Sieve of Eratosthenes

- **Purpose:** Find all primes up to $n$ in $O(n \log \log n)$.
- **Algorithm:**

1. Create a boolean array `isPrime` initialized to `true`.
2. Mark multiples of each prime starting from 2 as `false`.

**Code:**

```java
void sieve(int n) {
    boolean[] isPrime = new boolean[n + 1];
    Arrays.fill(isPrime, true);
    isPrime[0] = isPrime[1] = false;
    for (int i = 2; i * i <= n; i++) {
        if (isPrime[i]) {
            for (int j = i * i; j <= n; j += i) {
                isPrime[j] = false;
            }
        }
    }
}
```

---

### 3.2 Segmented Sieve

- **Purpose:** Efficiently find primes in a range $[L, R]$.
- Combines the Sieve of Eratosthenes with direct range checking.

---

### 3.3 Prime Factorization

- Use trial division up to $\sqrt{n}$.
- Efficient for multiple queries using a precomputed smallest prime factor (SPF) array.

---

## 4. Advanced Concepts

### 4.1 Euler's Totient Function (φ)

- **Definition:** Counts integers $k$ such that $1 \leq k \leq n$ and $\text{GCD}(k, n) = 1$.

- **Formula:**

$$\phi(n) = n \prod_{p|n} \left(1 - \frac{1}{p}\right)$$

where $p$ are prime factors of $n$.

---

### 4.2 Chinese Remainder Theorem

- Solves systems of congruences:

$$x \equiv a_1 \pmod{n_1}, \quad x \equiv a_2 \pmod{n_2}, \dots$$

- **Conditions:** $n_1, n_2, \dots$ are pairwise coprime.

---

### 4.3 Fermat's Little Theorem

- For a prime $p$:

$$a^p \equiv a \pmod{p}$$

If $a$ is not divisible by $p$:

$$a^{p-1} \equiv 1 \pmod{p}$$

---

## 5. Patterns and Applications

1. **Sum of Divisors:**
   - For $n = p_1^{e_1} \times p_2^{e_2}$:

$$\sigma(n) = (1 + p_1 + p_1^2 + \cdots + p_1^{e_1}) \times (1 + p_2 + \cdots + p_2^{e_2}) \dots$$

2. **Count of Divisors:**

$$d(n) = (e_1 + 1)(e_2 + 1) \dots$$

3. **Modulo Properties in Large Numbers:**

- Compute modulo directly at each step to avoid overflow.

---

Mastering these concepts will prepare you for challenging problems involving number theory during FAANG interviews. Let me know if you want example problems or further clarification!