Numerics / Math

A Gentle Introduction

A2SV
Africa To Silicon Valley

Outline

- Divisibility
- Prime numbers
- GCD

Divisibility

- We can describe any number N as the product of the **divisor d** and **the quotient q**, plus a **remainder r**.
    - N = q * d + r,   0 <= r < d
- Python syntax
    - **N % d == r**
    - **N // d == q**
    - **divmod(N, d) == (q, r)**, divmod is a function that returns (q, r)

Divisibility

- If `b % a == 0` then `(b * c) % a == 0`
- If `b % a == 0` then `(c * b) % (c * a) == 0`
- If `b % a == 0` and `c % b == 0`, then `c % a == 0`
- If `b % a == 0` and `c % a == 0`, then `(x*b + y*c) % a == 0`

Prime numbers

What are prime numbers?

2, 3, 5, 7, 13 ….

How to check if a number is prime?

How to check if a number is prime?

```python
def isPrime(x: int) -> bool:
    d = 2
    while d * d <= x:
        if x % d == 0:
            return False
        d += 1
    return True
```

Fundamental Theorem of Arithmetic

Theorem 4.3.1 (Fundamental Theorem of Arithmetic). *Every positive integer n can be written in a unique way as a product of primes:*

$$n = p_1 \cdot p_2 \cdots p_j \qquad (p_1 \le p_2 \le \cdots \le p_j)$$

Africa To Silicon Valley

Concrete Example Prime Fact.

Prime factorization of 84:

- $2^2 \times 3 \times 7$

Prime factorization of 52:

-

How to approach?

Prime factorization

```
def trial_division_simple(n: int) -> list[int]:
    factorization: list[int] = []
    d = 2

    while d * d <= n:
        while n % d == 0:
            factorization.append(d)
            n //= d
        d += 1
    if n > 1:
        factorization.append(n)

    return factorization
```

How does prime factorization relate to divisibility?

- If b % a == 0, then prime factors of a is a subset of prime factors of b.

How does prime factorization relate to divisibility?

Example: 24 is divisible by 4

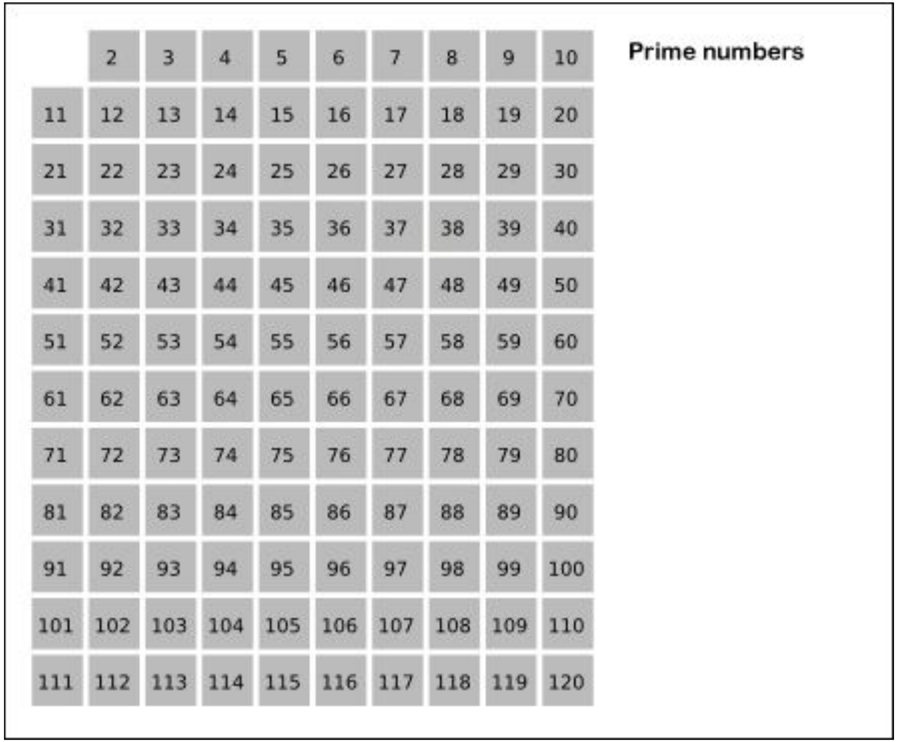
$4 = 2 * 2$

$24 = 2 * 2 * 2 * 3$

How to generate primes??

How to generate primes?

Sieve of Eratosthenes



A2SV
Africa To Silicon Valley

Sieve of Eratosthenes Implementation

```python
def prime_sieve(n: int) -> list[bool]:
    is_prime: list[bool] = [True for _ in range(n + 1)]
    is_prime[0] = is_prime[1] = False

    i = 2
    while i <= n:
        if is_prime[i]:
            j = 2 * i
            while j <= n:
                is_prime[j] = False
                j += i
        i += 1

    return is_prime
```

A2SV
Africa To Silicon Valley

How to optimize
Sieve of Eratosthenes?

Sieve of Eratosthenes

Optimizations

- Consider only proper multiples greater than square of the number
- Sieve till root
- Sieving by the odd numbers only

Sieve of Eratosthenes Optimized

```python
def prime_sieve(n: int) -> list[bool]:
    is_prime: list[bool] = [True for _ in range(n + 1)]
    is_prime[0] = is_prime[1] = False

    i = 2

    while i * i <= n:
        if is_prime[i]:
            j = i * i
            while j <= n:
                is_prime[j] = False
                j += i
        i += 1

    return is_prime
```

A2SV
Africa To Silicon Valley

Greatest Common Divisor (GCD)

- GCD of two numbers a and b is the greatest number that divides evenly into both a and b.
    - Naive algorithm
    - Euclidean algorithm (log(n))

Greatest Common Divisor (GCD)

How would you calculate gcd by hand?

Greatest Common Divisor (GCD)

In the prime factorisation method, each given number is written as the product of prime numbers and then find the product of the smallest power of each common prime factor.
This method is applicable only for positive numbers, i,e. Natural numbers.

Example: Find the Greatest common factor of 24, 30 and 36.

Solution: Prime factors of 24 is $2^3 \times 3$

Prime factors of 30 = $2 \times 3 \times 5$

Prime factors of 36 = $2^2 \times 3^2$

From the factorisation, we can see, only 2 x 3 are common prime factors.

Therefore, GCD (24, 30, 36) = 2 x 3 = 6

Exercise: Find the GCD of 32, 28

Greatest Common Divisor (GCD)

Greatest Common Divisor (GCD) Fast Algorithm

```
fun gcd(a, b):
    if b == 0:
        return a
    return gcd(b, a % b)
```

[Formal Proof]

Africa To Silicon Valley

Exercise: Find the GCD of 32, 28

Greatest Common Divisor (GCD) Fast Algorithm

Exercise

Complicated GCD

Count Primes

Divisibility by 2^n

"The enchanting charms of this sublime science reveal only to those who have the courage to go deeply into it."

~ Carl Friedrich Gauss

Resources for further reading

[Math for CS, MIT](#)
[CP-Algorithms](#)