Understanding the Factor Pairs Property

If a number n is **not prime**, it can be written as a product of two numbers:

n=a×bn = a \times bn=a×b

- If a is a divisor of n, then b = n / a is also a divisor.
- At least one of a or b **must** be **≤ sqrt(n)**.
- If a is greater than sqrt(n), then b must be smaller than sqrt(n).

This means we **only** need to check up to sqrt(n), because if n has a factor larger than sqrt(n), the corresponding factor **must be smaller**, and we would have already checked it.

Example: Checking n = 16

- The factors of 16 are:
  (1,16), (2,8), (4,4)
- The square root of 16 is sqrt(16) = 4.
- If 16 is not prime, it **must** have at least one factor ≤ 4.

Checking divisibility up to 4:

- 16 % 2 == 0 → Found a divisor! We can immediately say 16 is **not prime**.
- No need to check beyond 4 because 16 / 2 = 8 is already covered by the factor pair.

If we checked up to 15, we'd just be repeating unnecessary checks.

Efficiency Improvement

For a large number like 1,000,000, instead of checking **999,998** numbers (2 to 999,999), we only check up to **1,000** (sqrt(1,000,000) = 1000), making it **much faster**.

Final Takeaway

We only check **up to** sqrt(n) because:

- If n has a factor greater than sqrt(n), then the corresponding smaller factor was already checked.
- It eliminates unnecessary iterations, making the algorithm significantly faster.