Sieve of Eratosthenes:

**Linear sieve**

Intuition: if we cross out every composite only once, the algorithm will be linear.

Let q be a composite.

1. Initialize bool is\_prime[n] and vector<int> prime
2. For i = 2 to n
3. Check if is\_prime[i], if true add i to the vector
4. For j = 0; j < prime.size() and i \* prime[j] < n; j++
5. Is\_prime[i \* prime[j]] = false
6. If i % prime[j] == 0 break

**Multiplicative function**

Common multiplicative functions:

* Greatest common divisor of n and a constant
* Euler’s totient function
* Mobius function
* Number of divisors
* Sum of divisors
* Sum of a constant power of divisors

Example: (Euler’s totient function)

if :

if :

if :

if can’t think of the last case, can just use this: and maintain , the power of smallest prime factor of and maintain .