Multiplicative functions and Mobius Inversion

Prashant Shishodia

Indian Institute of Technology Kharagpur

2019

Contents

Motivation Problem
Problem

- Find the sum of gcd of all pairs of integers less than n. Formally, find,

$$\sum_{1 \le i,j \le n} gcd(i,j)$$

- dp solution ?

Motivation Problem
Problem

- Find the sum of gcd of all pairs of integers less than n. Formally, find,

$$\sum_{1 \le i,j \le n} gcd(i,j)$$

- dp solution ?

Motivation Problem
dp solution

- The first idea that strikes, is to consider $dp[i] =$ Number of pairs (x, y) such that$gcd(x, y) = i$
- Then, the integers less than n, divisible by i, is $\lfloor \frac{n}{i} \rfloor$ and therefore $dp_i = \lfloor \frac{n}{i} \rfloor^2$. But we have to subtract the pairs whose gcd is a multiple of d.

Motivation Problem
dp solution

- The first idea that strikes, is to consider $dp[i] =$ Number of pairs (x, y) such that$gcd(x, y) = i$
- Then, the integers less than n, divisible by i, is $\lfloor \frac{n}{i} \rfloor$ and therefore $dp_i = \lfloor \frac{n}{i} \rfloor^2$. But we have to subtract the pairs whose gcd is a multiple of d.

Motivation Problem
dp solution

- ```
  for(int i = n; i >= 1; --i){
      dp[i] = n/i * n/i;
      for(int j=2*i; j <= n; j += i)dp[i] -= dp[j];
  }
  ```
- Now, the answer to our problem will be $\sum_{i=1}^{n} i * dp[i]$
- Now, Let's see how we can solve this using mobius inversion.
  ```

Motivation Problem
dp solution

- ```
  for(int i = n; i >= 1; --i){
      dp[i] = n/i * n/i;
      for(int j=2*i; j <= n; j += i)dp[i] -= dp[j];
  }
  ```
- Now, the answer to our problem will be $\sum_{i=1}^{n} i * dp[i]$
- Now, Let's see how we can solve this using mobius inversion.
  ```

Motivation Problem

dp solution

- ```
  for(int i = n; i >= 1; --i){
      dp[i] = n/i * n/i;
      for(int j=2*i; j <= n; j += i)dp[i] -= dp[j];
  }
  ```
- Now, the answer to our problem will be $\sum_{i=1}^{n} i * dp[i]$
- Now, Let's see how we can solve this using mobius inversion.
  ```

Multiplication Functions

Definition

- A function $f : N- > C$ s is called multiplicative iff for any relatively prime integers m, n

$$f(mn) = f(m)f(n)$$

- Examples :
  - $\sigma(n)$: sum of divisors of n
  - $\tau(n)$: number of divisors of n

Multiplication Functions

Definition

- A function $f : N- > C$ s is called multiplicative iff for any relatively prime integers m, n

$$f(mn) = f(m)f(n)$$

- Examples :
  - $\sigma(n)$: sum of divisors of n
  - $\tau(n)$: number of divisors of n

Multiplication Functions
Properties

- Product of multiplicative functions is a multiplicative function
- Proof : Let $f(n)$ and $g(n)$ be multiplicative functions, and $h(n) = f(n)g(n)$, and m, n be coprime, then,

$$h(mn) = f(mn)g(mn) = f(m)f(n)g(m)g(n)$$

$$h(mn) = f(m)g(m) * f(n)g(n)$$

$$h(mn) = h(m)h(n)$$

Multiplication Functions
Properties

- Product of multiplicative functions is a multiplicative function
- Proof : Let $f(n)$ and $g(n)$ be multiplicative functions, and $h(n) = f(n)g(n)$, and m, n be coprime, then,

$$h(mn) = f(mn)g(mn) = f(m)f(n)g(m)g(n)$$

$$h(mn) = f(m)g(m) * f(n)g(n)$$

$$h(mn) = h(m)h(n)$$

Multiplication Functions

Properties

- If $f(n)$ be an multiplicative function then its sum function $S_f(n)$ defined as below is also multiplicative.

$$S_f(n) = \sum_{d|n} f(n)$$

- Proof : Let m, n be coprime, then

$$S_f(m)S_f(n) = (\sum_{d_1|n} f(d_1))(\sum_{d_2|m} f(d_2))$$

$$S_f(m)S_f(n) = \sum_{d_1|n \text{ and } d_2|m} f(d_1)f(d_2)$$

$$S_f(m)S_f(n) = \sum_{d_1*d_2|mn} f(d_1 * d_2)$$

$$S_f(m)S_f(n) = \sum f(d) = S_f(mn)$$

Multiplication Functions

Properties

- If $f(n)$ be an multiplicative function then its sum function $S_f(n)$ defined as below is also multiplicative.

$$S_f(n) = \sum_{d|n} f(n)$$

- Proof : Let m, n be coprime, then

$$S_f(m)S_f(n) = (\sum_{d_1|n} f(d_1))(\sum_{d_2|m} f(d_2))$$

$$S_f(m)S_f(n) = \sum_{d_1|n \text{ and } d_2|m} f(d_1)f(d_2)$$

$$S_f(m)S_f(n) = \sum_{d_1 * d_2|mn} f(d_1 * d_2)$$

$$S_f(m)S_f(n) = \sum f(d) = S_f(mn)$$

Multiplication Functions
Applications

- Doesn't make much sense? Don't worry, you will start connecting the dots once we discuss about mobius function.

Multiplication Functions

Applications

- So How exactly multiplicative functions make our life easier ? That's because they can be evaluated very easily than normal functions.
- How to evaluate multiplicative functions ? Let $f(n)$ be multiplicative function, and prime factorisation of n is $n = p_1^{r_1} p_2^{r_2} p_3^{r_3} ... p_k^{r_k}$

Multiplication Functions
Applications

- So How exactly multiplicative functions make our life easier ? That's because they can be evaluated very easily than normal functions.
- How to evaluate multiplicative functions ? Let $f(n)$ be multiplicative function, and prime factorisation of n is $n = p_1^{r_1} p_2^{r_2} p_3^{r_3} ... p_k^{r_k}$

Multiplication Functions
Application

- Then since all primes are coprime to each other.

$$f(n) = f(p_1^{r_1})f(p_2^{r_2})...f(p_k^{r_k})$$

- Thus we just need to compute the function at prime powers, and we can deduce answer for all other numbers from their prime factorisation.

Multiplication Functions

Application

- Then since all primes are coprime to each other.

$$f(n) = f(p_1^{r_1})f(p_2^{r_2})...f(p_k^{r_k})$$

- Thus we just need to compute the function at prime powers, and we can deduce answer for all other numbers from their prime factorisation.

Multiplication Functions
Problem

- Problem Link : Codeforces Problem
- Problem Statement : You have a number n written on board. You apply k operation on it. Suppose the number written on the board is v, then in one operation you will randomly choose one of its divisor and replace v with it. What is the expected value of number on board after k operations, modulo $10^9 + 7$.
- Constraints : $n \leq 10^{15}$, $k \leq 10^4$

Multiplication Functions
Problem

- Problem Link : Codeforces Problem
- Problem Statement : You have a number n written on board. You apply k operation on it. Suppose the number written on the board is v, then in one operation you will randomly choose one of its divisor and replace v with it. What is the expected value of number on board after k operations, modulo $10^9 + 7$.
- Constraints : $n \leq 10^{15}$, $k \leq 10^4$

Multiplication Functions
Solution

- Let's try a brute force dp solution first.
- Let $dp[n][k]$ =Expected number written on board after k turns, if we start at n, then $dp[n][0] = n$
- Then after one turn each divisor of n is chosen with equal probability($\tau(n)$ = number of divisors of n), hence we have.

$$dp[n][k] = \frac{1}{\tau(n)} \sum_{d|n} dp[d][k-1]$$

- But this will take $O(n * \tau(n) * k)$ time which is too slow.

Multiplication Functions

Solution

- Let's try a brute force dp solution first.
- Let $dp[n][k]$ =Expected number written on board after k turns, if we start at n, then $dp[n][0] = n$
- Then after one turn each divisor of n is chosen with equal probability($\tau(n)$ = number of divisors of n), hence we have.

$$dp[n][k] = \frac{1}{\tau(n)} \sum_{d|n} dp[d][k-1]$$

- But this will take $O(n * \tau(n) * k)$ time which is too slow.

Multiplication Functions

Solution

- Let's try a brute force dp solution first.
- Let $dp[n][k]$ =Expected number written on board after k turns, if we start at n, then $dp[n][0] = n$
- Then after one turn each divisor of n is chosen with equal probability($\tau(n)$ = number of divisors of n), hence we have.

$$dp[n][k] = \frac{1}{\tau(n)} \sum_{d|n} dp[d][k-1]$$

- But this will take $O(n * \tau(n) * k)$ time which is too slow.

Multiplication Functions

Use of Multiplicative function

- So let's use properties of multiplicative functions to get better time complexity.
- Observation : Note that $f(n) = dp[n][k]$ is multiplicative function for given k. But how do we prove it ?

Multiplication Functions

Use of Multiplicative function

- So let's use properties of multiplicative functions to get better time complexity.
- Observation : Note that $f(n) = dp[n][k]$ is multiplicative function for given k. But how do we prove it ?

Multiplication Functions

Use of Multiplicative function

- We can try proving with Induction on k. Note that for $k = 0$ $dp[n][k] = n$ is multiplicative.
- Now assume that $dp[n][k-1]$ is multiplicative, then both functions $\frac{1}{\tau(n)}$ and $\sum_{d|n} dp[d][k-1]$ is multiplicative, and hence their product is also multiplicative.

Multiplication Functions

Use of Multiplicative function

- We can try proving with Induction on k. Note that for $k = 0$ $dp[n][k] = n$ is multiplicative.
- Now assume that $dp[n][k-1]$ is multiplicative, then both functions $\frac{1}{\tau(n)}$ and $\sum_{d|n} dp[d][k-1]$ is multiplicative, and hence their product is also multiplicative.

Multiplication Functions

Use of Multiplicative function

- Thus we just need to calculate our dp on prime powers that divide n. And since there are atmost $\log n$ prime powers. Time complexity is $O(\sqrt{n} + klogn)$. ($\sqrt{n}$ term for prime factorisation).
- Formally, consider dp as $dp[p][j][k]$ = Expected number on board after k turns if we start with p^j, and if $n = p_1^{r_1} p_2^{r_2} ... p_k^{r_k}$, we have,

$$dp[p][j][k] = \frac{1}{j+1} \sum_{j_1=0}^{j} dp[p][j_1][k-1]$$

$$dp[n][k] = \Pi_{i=1}^{k} dp[p_i][r_i][k]$$

Multiplication Functions

Use of Multiplicative function

- Thus we just need to calculate our dp on prime powers that divide n. And since there are atmost $\log n$ prime powers. Time complexity is $O(\sqrt{n} + klogn)$. ($\sqrt{n}$ term for prime factorisation).
- Formally, consider dp as $dp[p][j][k]$ = Expected number on board after k turns if we start with p^j, and if $n = p_1^{r_1} p_2^{r_2} ... p_k^{r_k}$, we have,

$$dp[p][j][k] = \frac{1}{j+1} \sum_{j_1=0}^{j} dp[p][j_1][k-1]$$

$$dp[n][k] = \Pi_{i=1}^{k} dp[p_i][r_i][k]$$

Mobius Inversion
Properties

- Mobius function $\mu(n)$ is defined as

$$\mu(n) = \begin{cases} (-1)^r, & \text{if } n = p_1 p_2 p_3 ... p_r \\ 0, & \text{otherwise} \end{cases}$$

- Intuitively, $\mu(n)$ is 0 for non - squarefree integers (which are divisible by square of some prime), and -1 or 1 depending on if the number of prime factors of n is odd or even. Note than $\mu(n)$ is multiplicative function.
- We also consider the simplest of functions (not so simple in application :-)), dirchlet function $e(n)$, as

$$e(n) = \begin{cases} 1, & \text{iff } n = 1 \\ 0, & \text{otherwise} \end{cases}$$

Mobius Inversion

Properties

- Mobius function $\mu(n)$ is defined as

$$\mu(n) = \begin{cases} (-1)^r, & \text{if } n = p_1 p_2 p_3 ... p_r \\ 0, & \text{otherwise} \end{cases}$$

- Intuitively, $\mu(n)$ is 0 for non - squarefree integers (which are divisible by square of some prime), and -1 or 1 depending on if the number of prime factors of n is odd or even. Note than $\mu(n)$ is multiplicative function.
- We also consider the simplest of functions (not so simple in application :-)), dirchlet function $e(n)$, as

$$e(n) = \begin{cases} 1, & \text{iff } n = 1 \\ 0, & \text{otherwise} \end{cases}$$

Mobius Inversion

Properties

- Mobius function $\mu(n)$ is defined as

$$\mu(n) = \begin{cases} (-1)^r, & \text{if } n = p_1 p_2 p_3 ... p_r \\ 0, & \text{otherwise} \end{cases}$$

- Intuitively, $\mu(n)$ is 0 for non - squarefree integers (which are divisible by square of some prime), and -1 or 1 depending on if the number of prime factors of n is odd or even. Note than $\mu(n)$ is multiplicative function.
- We also consider the simplest of functions (not so simple in application :-)), dirchlet function $e(n)$, as

$$e(n) = \begin{cases} 1, & \text{iff } n = 1 \\ 0, & \text{otherwise} \end{cases}$$

Mobius Inversion
Properties

- Now consider sum function of $\mu(n)$

$$S_\mu(n) = \sum_{d|n} \mu(d) = e(n)$$

- Proof (intuitive) : $n = 1$ is trivial, for $n \geq 1$ Suppose n has r distinct primes in its factorisation, then product of primes in each subset will contribute to $S_\mu(n)$, and since, half of them will contain odd primes, and half even, they will cancel each other, hence $S_\mu(n) = 0$

Mobius Inversion

Properties

- Now consider sum function of $\mu(n)$

$$S_\mu(n) = \sum_{d|n} \mu(d) = e(n)$$

- Proof (intuitive) : $n = 1$ is trivial, for $n \geq 1$ Suppose n has r distinct primes in its factorisation, then product of primes in each subset will contribute to $S_\mu(n)$, and since, half of them will contain odd primes, and half even, they will cancel each other, hence $S_\mu(n) = 0$

Mobius Inversion

Properties

- Proof (Formal) : Let $n(>1) = p_1^{r_1}...p_k^{r_k}$, then ($S_\mu$ is multiplicative since μ is multiplicative)

$$S_\mu = \Pi_{i=1}^{k} S_\mu(p_i^{r_i})$$

$$S_\mu = \Pi_{i=1}^{k}(1 + \mu(p_i) + \mu(p_i^2)..\mu(p_i^{r_i}))$$

$$S_\mu = \Pi_{i=1}^{k}(1 - 1 + 0... + 0) = 0$$

Mobius Inversion

Mobius Inversion

- So now, what the heck is Mobius inversion ? It's nothing but the simple identity, that we've already discussed.

$$e(n) = \sum_{d|n} \mu(d)$$

- It's okay if you're feeling like, "Seriously ? are you not kidding me ? I read all this for this poor equation that doesn't seem to posses any superpowers ? ". Don't worry You'll see the magic in seconds.

Mobius Inversion

Mobius Inversion

- So now, what the heck is Mobius inversion ? It's nothing but the simple identity, that we've already discussed.

$$e(n) = \sum_{d|n} \mu(d)$$

- It's okay if you're feeling like, "Seriously ? are you not kidding me ? I read all this for this poor equation that doesn't seem to posses any superpowers ? ". Don't worry You'll see the magic in seconds.

Mobius Inversion

Magic

- We will use [some condition] to denote 1 if condition is true and 0 otherwise.

Mobius Inversion
Magic

- Now, lets solve an easy problem : Find the pair of coprime intgers not more than n. Formally, find,

$$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} [gcd(i,j) == 1]$$

$$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} e[gcd(i,j)]$$

- Now, the magic, apply mobius inversion.

$$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{d|gcd(i,j)} \mu(d)$$

Mobius Inversion

Magic

- Now, lets solve an easy problem : Find the pair of coprime intgers not more than n. Formally, find,

$$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} [gcd(i,j) == 1]$$

$$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} e[gcd(i,j)]$$

- Now, the magic, apply mobius inversion.

$$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{d|gcd(i,j)} \mu(d)$$

Mobius Inversion
The Black Magic

- That gives,

$$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{d=1}^{n} \mu(d) * [d | gcd(i,j)]$$

-

$$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{d=1}^{n} \mu(d) * [d | i \text{ and } d | j]$$

-

$$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{d=1}^{n} \mu(d) * [d | i] * [d | j]$$

Mobius Inversion

The Black Magic

- That gives,

$$f(n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{d=1}^{n}\mu(d) * [d|gcd(i,j)]$$

- $$f(n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{d=1}^{n}\mu(d) * [d|i \text{ and } d|j]$$

- $$f(n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{d=1}^{n}\mu(d) * [d|i] * [d|j]$$

Mobius Inversion
The Black Magic

- That gives,

$$f(n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{d=1}^{n} \mu(d) * [d|gcd(i,j)]$$

- $$f(n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{d=1}^{n} \mu(d) * [d|i \text{ and } d|j]$$

- $$f(n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{d=1}^{n} \mu(d) * [d|i] * [d|j]$$

Mobius Inversion

The Black Magic

- Rearranging,

$$f(n) = \sum_{d=1}^{n} \mu(d) * (\sum_{i=1}^{n} [d|i]) * (\sum_{j=1}^{n} [d|j])$$

- But , $\sum_{i=1}^{n} [d|i] = \lfloor \frac{n}{d} \rfloor$
- Hence,

$$f(n) = \sum_{d=1}^{n} \mu(d) * \lfloor \frac{n}{d} \rfloor^2$$

Mobius Inversion

The Black Magic

- Rearranging,

$$f(n) = \sum_{d=1}^{n} \mu(d) * (\sum_{i=1}^{n} [d|i]) * (\sum_{j=1}^{n} [d|j])$$

- But , $\sum_{i=1}^{n} [d|i] = \lfloor \frac{n}{d} \rfloor$
- Hence,

$$f(n) = \sum_{d=1}^{n} \mu(d) * \lfloor \frac{n}{d} \rfloor^2$$

Mobius Inversion
The Black Magic

- Rearranging,

$$f(n) = \sum_{d=1}^{n} \mu(d) * (\sum_{i=1}^{n} [d|i]) * (\sum_{j=1}^{n} [d|j])$$

- But , $\sum_{i=1}^{n} [d|i] = \lfloor \frac{n}{d} \rfloor$
- Hence,

$$f(n) = \sum_{d=1}^{n} \mu(d) * \lfloor \frac{n}{d} \rfloor^{2}$$

Mobius Inversion

Computation ?

- Now, how to calcualte $f(n)$?
- $\mu(n)$ can be calculate in linear or $O(nloglogn)$ time using sieve, one implementation is,

```
mu[n], is_prime[n]   // {initialize mu, is_prime to 1
for(int i=2;i<=n;++i)if(is_prime[i]){
    for(int j=i;j<n;j += i){
        if(j > i)is_prime[j] = 0
        if(j%(i*i) == 0)mu[j] = 0
        mu[j] = -mu[j]
    }
}

f(n) = 0
for(int d=1;d<=n;++d)
    f(n) += mu(d) * n/d * n/d
```

Mobius Inversion
Computation ?

- Now, how to calcualte $f(n)$?
- $\mu(n)$ can be calculate in linear or $O(nloglogn)$ time using sieve, one implementation is,

```
mu[n], is_prime[n]   // {initialize mu, is_prime to 1
for(int i=2;i<=n;++i)if(is_prime[i]){
    for(int j=i;j<n;j += i){
        if(j > i)is_prime[j] = 0
        if(j%(i*i) == 0)mu[j] = 0
        mu[j] = -mu[j]
    }
}

f(n) = 0
for(int d=1;d<=n;++d)
    f(n) += mu(d) * n/d * n/d
```

Mobius Inversion
Original Problem

- Let's come back to the original problem. Find

$$g(n) = \sum_{1 \le i,j \le n} gcd(i,j)$$

- g(n) =

$$\sum_{1 \le i,j \le n} gcd(i,j)$$

$$g(n) = \sum_{d=1}^{n} d * \sum_{i=1}^{n} \sum_{j=1}^{n} [gcd(i,j) == d]$$

$$g(n) = \sum_{d=1}^{n} d * \sum_{i=1}^{n} \sum_{j=1}^{n} [gcd(\frac{i}{d}, \frac{j}{d}) == 1]$$

$$g(n) = \sum^{n} d * \sum^{\frac{n}{d}} \sum^{\frac{n}{d}} [gcd(i,j) == 1]$$

Mobius Inversion

Original Problem

- But we have already seen

$$\sum_{i=1}^{\frac{n}{d}} \sum_{j=1}^{\frac{n}{d}} [gcd(i,j) == 1]$$

- Thus,

$$g(n) = \sum_{d=1}^{n} d * f(\lfloor \frac{n}{d} \rfloor)$$

- Thus, $g(n)$ can be calculated by calculating $f(n)$ from above in linear time.

Mobius Inversion
Original Problem

- But we have already seen

$$\sum_{i=1}^{\frac{n}{d}} \sum_{j=1}^{\frac{n}{d}} [gcd(i,j) == 1]$$

- Thus,

$$g(n) = \sum_{d=1}^{n} d * f(\lfloor \frac{n}{d} \rfloor)$$

- Thus, $g(n)$ can be calculated by calculating $f(n)$ from above in linear time.

Mobius Inversion
Original Problem

- But we have already seen

$$\sum_{i=1}^{\frac{n}{d}} \sum_{j=1}^{\frac{n}{d}} [gcd(i,j) == 1]$$

- Thus,

$$g(n) = \sum_{d=1}^{n} d * f(\lfloor \frac{n}{d} \rfloor)$$

- Thus, $g(n)$ can be calculated by calculating $f(n)$ from above in linear time.

Mobius Inversion

References and Problems

- A dance with Mobius function
- Math Node : Mobius Inversion
- Multiplicative functions

Mobius Inversion
Original Problem

That's all folks. Practice more, Keep improving.