
Mobius Function

Mobius Function is a function $\mu()$, where:

- $\mu(x) = 0$, if x is divisible by some perfect square (except 1).
- $\mu(x) = 1$, if it is square free and contains even number of prime factors.
- $\mu(x) = -1$, if it is square free and contains odd number of prime factors.

It is mostly used in problems that require inclusion-exclusion. In fact, the function is simply a tool to make inclusion-exclusion simpler to apply.

Generating Mobius Function

Suppose we want to generate mobius function till n. Then do the following:

- Generate prime numbers till $\sqrt{n}$.
- Declare an array of size n, let assign 1 to every position. Let this array be `mu[]`.
- For every prime number p, mark every multiple of p^2 as 0 in `mu[]`.
- Then for every prime p multiply its multiple with -1.

```
vector<int> prime; ///Already generated using sieve
int mu[SIZE];
void calculateMobiusFunction( int n ) {
  for ( int i = 1; i <= n; i++ ) mu[i] = 1;
  int sqrtn = sqrt ( SIZE );

  /*Remove the numbers with squares*/
  for ( int i = 0; i < prime.size(); i++ ){
    if ( prime[i] > sqrtn ) break;
    int x = prime[i] * prime[i];

    /*Mark every multiple of x as 0, cause they are divisible by prime^2*/
    for ( int j = x; j <= n; j += x ) mobius[j] = 0;
  }

  for ( int i = 0; i < prime.size(); i++ ){

    /*Multiply every multiple of prime[i] with -1*/
    for ( int j = prime[i]; j <= SIZE; j += prime[i] ) {
      mobius[j] *= -1;
    }
  }
}
```

When to Use Mobius Function?

Whenever we have to do sion-exclusion using the first k primes, using mobius function makes things easier for us.

Application

Counting numbers that cannot be expressed as exponent

Given the value N, find count of number x, where $x \leq N$ and it cannot be expressed as a^b (where $b > 1$).

We can use inclusion-exclusion augmented with mobius function to solve the problem.

Initially, let $res = N$. From the result, let us exclude all numbers that can be written as a^2. Let such numbers be $x_2 = \sqrt[2]{N}$. Similarly, let us exclude $x_3 = \sqrt[3]{N}$, since they can be written as a^3. We don't need to exclude x_4 since they were removed by x_2. Next we exclude x_5. Next comes x_6. Now notice that numbers in x_6 were removed twice: once by x_2 and once by x_3. Hence, we add x_6.

What gets added, removed or ignored is decided my mobius function here.