NSUPS Bootcamp Week 3

Getting more and more serious

Number Theory

We will see some more cool number theory topics

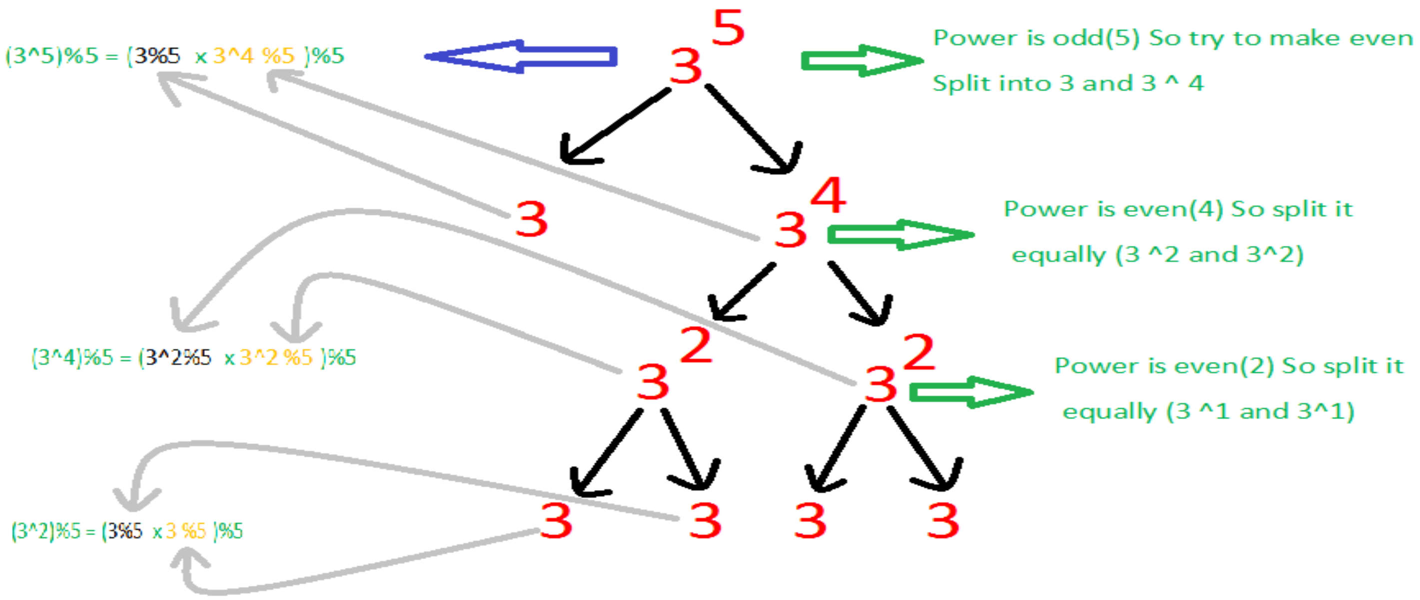
1. Power of a^b
2. Big Mod (a^b)%mod
3. Prime Factorization
4. Number of divisors
5. Sum of divisors

Power (a^b)

1. Naive approach is to iterate from 1 to a.
2. But we can improve it to logN using a simple trick.
3. Check the picture in the next slide to visualize the approach.

Big Mod (a^b)%mod

1. The same approached we used in power.
2. We just need to mod the result each time we do any arithmetic operation.



Sample Code of Big Mod

```
// Calculate (a^p)%m
 long long  bigmod ( long long a, int p, int m )
 {
     if ( p == 0 ) return 1; // If power is 0, then a ^ 0 = 1 for any value of a, And 1 Mod m=1
 for any value of m, So return 1

     if ( p % 2 ) // If power is odd, Split it : a ^ 5 =( a )* (a ^ 4) --> left and right child
 respectively.
     {
         return ( ( a % m ) * ( bigmod ( a, p - 1, m ) ) ) % m;
     }
     else //If power is even then split it equally and return the result...
     {
         long long c = bigmod ( a, p / 2, m ); /// Both part will have the same result
         return ( (c%m) * (c%m) ) % m;
     }
 }
```

Prime Factorization

Given a number N we want to find its prime factorization, prime factorization of 60 = 2*2*3*5.

1. Iterate for number p from 2 to N and find how many times it divides N.
2. We don’t need to check for any p greater than sqrt(N).
3. If more than 1 prime factor p > sqrt(N) exist, then p*p>N which is impossible.
4. When p divides N then make it smaller by dividing it by N and update sqrt(N) accordingly to make it more faster.
5. We don’t need to test for each number till sqrt(N). We are searching for the prime factors, so loop through each prime till sqrt(N).

Application of prime factorization

1. Find LCM and GCD.
2. Find number of trailing zeros in N!.
3. Find number of divisors and sum of divisors (check later slides).
4. Find nCr for large numbers.

Sample Code of Prime Factorization

```
vector<int>prime; // Contains all the prime numbers from 1 to n
vector<int>factors; /// After factorize() it’ll contain the prime factors of n

void factorize( int n ) {
    int sqrtn = sqrt ( n );
    for ( int i = 0; i < prime.size() && prime[i] <= sqrtn; i++ ) {
        if ( n % prime[i] == 0 ) { // Found a prime that divides n
            while ( n % prime[i] == 0 ) { // Check how many times it divides n
                n /= prime[i];
                factors.push_back(prime[i]);
            }
            sqrtn = sqrt ( n );
        }
    }
    if ( n != 1 ) {
        factors.push_back(n); // The only prime factor > sqrt(n)
    }
}
```

Number of Divisors

1. We can apply the same approach we did for finding the divisors.
2. But we can do better using prime factorization.
3. Let’s take a number x = 180 whose prime factorization is 2*2*3*3*5.
4. Let’s divide the prime factors into 3 sets.
5. S1 = [2^0, 2^1, 2^2], S2 = [3^0, 3^1, 3^2], S3 = [5^0 , 5^1]
6. Now let’s say d is a divisor of x. In how many ways we can form a d?
7. Important thing to notice here is, d must be a multiplication of a*b*c where a is an element from S1, b is from S2 and c is from S3.
8. In how many ways we can select a, b and c? It is size(S1)*size(S2)*size(S3).
9. So the number of divisors of 180 is 4*4*3.

Sum of Divisors

1. It’s very similar with our previous approach.
2. We can apply the formula of infinite series.
3. Take the same example from previous slide.
4. Sum of divisors of 180 = (2^0 + 2^1 + 2^2)*(3^0 + 3^1 + 3^2)*(5^0 + 5^1).

Number of Digits in N!

1. How do you find the number of digits in a number?

2. Number of digits in N = $\lfloor log_{10} N \rfloor$ + 1
   Warning: Be careful about precision error!

Then, how do we calculate the number of digits in N! ?

How many digits does N have in base 15?

More on Factorials

1. Find K **trailing** non-zero digit of N!
2. Find K **leading** digits of N!
3. What about factorials in different **base**?

Resources

1. Prime Factorization http://forthright48.blogspot.com/2015/07/prime-factorization-of-integer.html
2. Number of Divisors http://forthright48.blogspot.com/2015/07/number-of-divisors-of-integer.html
3. Sum of Divisors http://forthright48.blogspot.com/2015/07/sum-of-divisors-of-integer.html
4. Number of digits in N! https://forthright48.blogspot.com/2015/08/number-of-digits-of-factorial.html
5. Leading K digits of N! https://forthright48.blogspot.com/2015/08/leading-digits-of-factorial.html