1 Math

1.1 POJ 2115

According to the probelm description, we are looking for an x, such that

$$A + Cx \equiv B(mod\ 2^k)$$

then we can get

$$Cx \equiv B - A(mod\ 2^k)$$

from the Congruence equation quality, we will transfrom it into an equation

$$2^k x + Cy = B - A$$

is available, where x and y is what we are looking for. Now we turn to find the answer of

$$2^k x_0 + Cy_0 = gcd(2^k, C)$$

that is completely the Euclid form. We will use Extended Euclid Algorithm to solve this function. As we get the answer of $y0$ by recurring, we will try to get y through

$$y = y_0 * \frac{B - A}{gcd(a, b)}$$

but for this problem, we must find the least positive one. Let x_1, y_1 satisfy:

$$2^k x_1 + Cy_1 = gcd(2^k, C)$$

Observing these two equations, we can get:

$$2^k(x_0 - x_1) = C(y_1 - y_0)$$

and divided by $gcd(2^k, C)$ both side:

$$\frac{2^k(x_0 - x_1)}{gcd(2^k, C)} = \frac{C(y_1 - y_0)}{gcd(2^k, C)}$$

where $x_0 - x_1$ is a multiple of $\frac{C}{gcd(2^k,C)}$, correspondently $y_1 - y_0$ is a multiple of $\frac{2^k}{gcd(2^k,C)}$, for 2^k is prime to $gcd(2^k, C)$, and same to C. Thus we can conclude an significant equation :
if we have $ax + by = c$:

$$\begin{cases} x = x' + k\dfrac{b}{gcd(a,b)} \\ y = y' - k\dfrac{a}{gcd(a,b)} \end{cases}, k \in Z$$

so that we can find **the least positive one** matched to the problem, **(Noted that the equation above is irrelevent to the right constant!)**

$$\begin{cases} x = (x\%\dfrac{b}{gcd(a,b)} + \dfrac{b}{gcd(a,b)})\%\dfrac{b}{gcd(a,b)} \\ y = (y\%\dfrac{a}{gcd(a,b)} + \dfrac{a}{gcd(a,b)})\%\dfrac{a}{gcd(a,b)} \end{cases}$$

such we call **the least positive equation**
And then we will use **the least positive equation** to find the answer of the problem finally. Here we can prove the Extended Euclid Algorithm: let

$$\begin{cases} ax_1 + by_1 = gcd(a\ b) \\ bx_2 + (a\%b)y_2 = gcd(b, a\%b) \end{cases}, a > b > 0$$

According to the Euclid pricipal, $gcd(a\ b) = gcd(b, a\%b)$. we can get:

$$\begin{aligned} ax_1 + by_1 &= bx_2 + (a\%b)y_2 \\ &= bx_2 + (a - a/b * b)y_2 \\ &= ay_2 + bx_2 - a/b * by_2 \\ &= ay_2 + b(x_2 - a/b * y_2) \end{aligned}$$

relatively,

$$\begin{cases} x_1 = y_2 \\ y_1 = x_2 - a/b * y_2 \end{cases}$$

which is the recursion we need in recurring function of Extended Euclid algorithm.

1.2 POJ 2115

we are to calculate

$$sum = \sum_{i=1}^{N} gcd(N, i)$$

take N=6 as an example:

i	1	2	3	4	5	6
gcd(i,6)	1	2	3	2	1	6

we take gcd(i,N) as g_i we can know g_i appears exactly $\phi(\frac{N}{g_i})$ times, where ϕ is Euler function. because of:

$$(\frac{N}{g_i}, \frac{i}{g_i}) = 1$$

and the meaning of $\phi(x)$ is the number of figure prime to the x.
Thus $\phi\frac{N}{g_i})$ corresponds to the number of $\frac{i}{g_i}$, which is also the number of i, such that $gcd(N, i) = g_i$
On the other hand, if we take i, which is the factor of N, then $gcd(N, i)$ is exactly i, and the number of which we can calculated is $\phi(\frac{N}{g_i} = \frac{N}{i})$
Therefore, traversing i range from 1 to N to find the factor of N is the essential optimization to the algorithm.