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.

1.3 HDU 6425

It bases on the inclusive-exclusive priciple.
a is the number of people have neither rackets nor balls, so it is unnecessary to include them to calculate "orgnizing fail", that is under

the condition that at least one ball and at least two rackets inclusive. So 2^a means none of students for the number b,c,d is inclusive.
Then we exclude these group now.
Similarly, 2^b is the number of students who don't have any ball no matter they are labeled c or d. 2^c is the number of students who don't have any racket no matter they are labeled b or d, while $2^c(b+d)$ worth our attention, for it represents the students with one ball and one racket, which is obviously the condition that meet the problem request.
According to the inclusive-exclusive priciple, we have got the the number of rackets failing ad balls failing, so the next target is find not only rackets but balls fail to meet request, that is with no ball and one-or-zero rackets.
The number of which is $b+1$ so the totally sum is

$$2^a(2^b + (b+d+1)2^c - (b+1))$$

It is noteworthy of the modular operating.

1.4 HDU 6432

考虑使用容斥原理进行计数. 包含至少一个形如 $[i, i+1]$ 或 $[n, 1]$ 这样的子串的环排列个数是 $\binom{n}{1}(n-2)!$ 个
可以推广为包含至少 $k(0 \le k < n)$ 个的环排列个数是

$$\binom{n}{k}(n-k-1)!$$

同时注意到包含 n 个的环排列个数一定是 1 个.
所以最终答案就是

$$(-1)^n + \sum_{k=0}^{n-1}(-1)^k\binom{n}{k}(n-k-1)!$$

1.5 HDU 6434

令 $a = i - j$, 先枚举 i 再枚举 a

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{i-1}[\gcd(i+j, i-j) = 1] \\
=&\sum_{i=1}^{n}\sum_{a=1}^{i-1}[\gcd(2i-a, a) = 1] \\
=&\sum_{i=1}^{n}\sum_{a=1}^{i-1}[\gcd(2i, a) = 1]
\end{aligned}$$

即对于每个 i, 求有多少个小于它的 a 满足 $\gcd(i, a) = 1$ 且 a 是奇数.
当 i 是奇数时, 答案为 $\frac{\varphi(i)}{2}$.
当 i 是偶数时, 答案为 $\varphi(i)$.
注意 $i = 1$ 时, 答案为 0.
记个前缀和就好了, 复杂度为 $O(N + T)$.