Competitive programming
- Elydeck

→ Prime sieve
Sieve of erathestenes

Make an array (bool) of size n.
Mark all true.

T	T	T	T	T	T	T	T	T	T	T
0	1	2	3	4	5	6	7	8	9	10

$A[0]$ = false; $A[1]$ = false;

We need to find all primes from 2 to n.

```
for (i = 2; i <= n; i++)
{
    if (A[i] == true) {
        for (j = i*i; j <= n; j += i)
        {
            A[j] = false;
        }
    }
}
```

print those index which are marked true.

Time - complexity = $n \log \log n$

Finding GCD

Euclid's algorithm

$$gcd(a,b) = gcd(b, a\%b)$$

when $a > b$

Learn it as "bab"

$$gcd(a, 0) = a$$

→ Extended - euclid's algorithm

$$gcd(a,b) = ax + by$$

we need to find x & y.

$$ax + by = gcd(a,b) \qquad \underline{\quad} \ i)$$

Also, $\quad gcd(a,b) = gcd(b, a\%b) \qquad \underline{\quad} ii)$

From i) & ii)

$$ax + by = gcd(b, a\%b) \qquad \underline{\quad} iii)$$

→ $\quad bx_1 + (a\%b)y_1 = gcd(a,b) = ax + by$

→ $\quad bx_1 + \left\{ a - \left[\dfrac{a}{b}\right]\left[\dfrac{a}{b}\right]\cdot b \right\} y_1 = ax + by$

→ On comparing, we get

$$\boxed{x = y_1} \quad , \quad \boxed{y = x_1 - \left[\dfrac{a}{b}\right]y_1}$$

→ Number of balanced Binary Tree
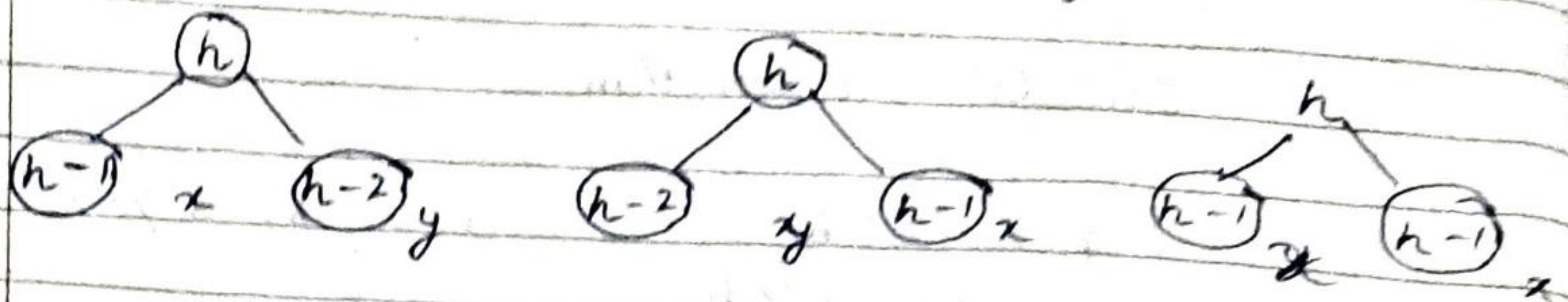
$$ans = 2x + y$$

```
int balanced BT (int h) {
    if (h==0 || h==1) {
        return 1;
    }
    int m = pow(10, 9) + 7;
    int x = balanceBT (h-1);
    int y = balanceBT (h-2);
    long res1 = (long) x * x;
    long res2 = (long) x * y * 2;
    int ans1 = (int) (res1 % m);
    int ans2 = (int) (res2 % m);
    return (ans1 + ans2) % m;
}
```

$$Balance = |LH - RH| \leq 1$$

$$(h-1) \longrightarrow x$$
$$(h-2) \longrightarrow y$$

$$ans = x * x + x * y + x * y$$
$$= x^2 + 2xy$$

modulo inverse &

$$a \times b = 1$$
$$b = \frac{1}{a} \quad \text{(multiplicative inverse)}$$

Multiplicative mod inverse

$$(A \cdot B) \% m = 1$$

We have to find B.

$$(A \cdot B) \% m = 1$$

$$\Rightarrow ((A \% B) \cdot (B \% m)) \% m = 1$$

$$\Rightarrow \therefore (a \cdot b) \% m = ((a \% m) \cdot (b \% m)) \% m$$

$$\Rightarrow 1 \leq B \leq m-1$$

$$(a \cdot b) \% m = 1$$

$$\Rightarrow (a \cdot b) \% m - 1 = 0$$

$$a \cdot b \equiv 1$$

$$\Rightarrow a \cdot b - 1 \equiv 0 \quad \text{(multiple of m)}$$

$$= a \cdot b - 1 \equiv mq$$

$$= a \cdot b + mQ = 1$$

$$= a \cdot b + mQ = 1$$

$$\downarrow \qquad = gcd(a, m) = 1$$

$$(ax + by) = gcd(a, b)$$

mod inverse (a, m)

$$= b$$

matlab a me kya multiply kre
aur m se mod kre ki ans = 1.

mod inverse (a, m) = b

$$\Rightarrow (a \times ?) \% m = 1$$
$$\uparrow$$

we have to find this only

→ Hard question

$$ax + by = d$$

a & b item se d banana hai
Need to find number of pairs
possible of (x, y) for such

$$ax + by = d$$

$$ax = \underbrace{d - by}$$

This term must be
divisible by x.

$$0 \le y \le \frac{d}{b} , \qquad 0 < x \le \frac{d}{a}$$

$$ax = d - \underset{\downarrow}{by}$$

let us assume y_1 to be minimum value for such condition to be hold.

$$d \rightarrow \text{fixed}$$
$$b \rightarrow \text{fixed}$$
$$y \rightarrow \text{can change.}$$

$$ax = d - by_1$$

Next term, $\quad d - b(y_1 + a)$
$$d - b(y_1 + 2a)$$
$$d - b(y_1 + 3a)$$
$$\vdots$$
$$d - b(y_1 + na)$$

On comparing with maximum value of y

$$y_1 + na = \frac{d}{b}$$

$$na = \frac{d}{b} - y_1$$

$$\boxed{n = \frac{\frac{d}{b} - y_1}{a}} \qquad \begin{array}{l} \text{Total terms} \\ = 1 + n \end{array}$$

Now we need to find y_1.

$$(d - by_1) \% a = 0$$

$$\Rightarrow \quad d \% a - (by_1) \% a = 0$$

$$\Rightarrow \quad d \% a - [(b \% a) \cdot (y_1 \% a)] \% a = 0 \quad \text{—i)}$$

$$ax + by = d$$

$$y = \frac{d - ax}{b}$$

For smallest value,

$$y_1 \% a = \left(\frac{d - ax}{b}\right) \% a$$

$$= \left(\frac{d}{b}\right) \% a - \frac{(ax)}{b} \% a$$

$$y_1 \% a = \left(\frac{d}{b}\right) \% a \quad \text{—ii)}$$

i) & ii)

$$\Rightarrow \quad d \% a - \left[[b \% a] \cdot \left(\frac{d}{b}\right) \% a \right] \% a = 0$$

$$\Rightarrow \quad d \% a - \left[\left(\frac{b \cdot d}{b}\right) \% a \right] \% a = 0$$

$$d \% a - (d \% a) \% a = 0$$

this holds

$$y_1 = \left(\frac{d}{b}\right) \% a$$

$$= d^* \text{ mod inverse } (b, a)$$

```
    g = gcd (a,b);
if (d % g == 1) {
        cout << 0 << endl;
    }
if (d == 0) {
        cout << 1 << endl;
    }
a / = g;
b / = g;
d / = g;
long value = ((d % a) * mod Inverse (b,a)) % a;
long first value = d / b
if (d < y1 * b) {
        cout << 0 << endl;
    }
else {
    long n = (first value - y1) / a;
    cout << 1 + n << endl;
    }
```

$$y_1 = \left(\frac{d}{b}\right) \% \ a$$

$$ax + by = d$$

$$\frac{ax}{g} + \frac{by}{g} = \frac{d}{g}$$

$$Ax + By = D$$

→ Advanced GCD

$$gcd(a, b) = gcd(b, a \% b)$$

both must be integers.

$$gcd(10^{240}, 40) = gcd(40, 10^{240} \% 40)$$

Input as string.

ex— $(2\ 3\ 5\ 6\ 7) \% 40$

$= (0^* 10 + 2) \% 40 = 2$

$(2^* 10 + 3) \% 40 = 23$

$(23^* 10 + 5) \% 40 =$

$2356 \% 40 = (235^* 10 + 6) \% 40$

$= (235^* 10) \% 6 + 6 \% 40) \% 40$

$= (from\ previous + 6 \% 40) \% 40$

```
ll gcdLarge (ll a, string b) {
    ll mod = 0;
    for (int i = 0; i < b.size(); i++) {
        mod = (mod * 10 + b[i] - '0') % a;
    }
    return mod;
}
```
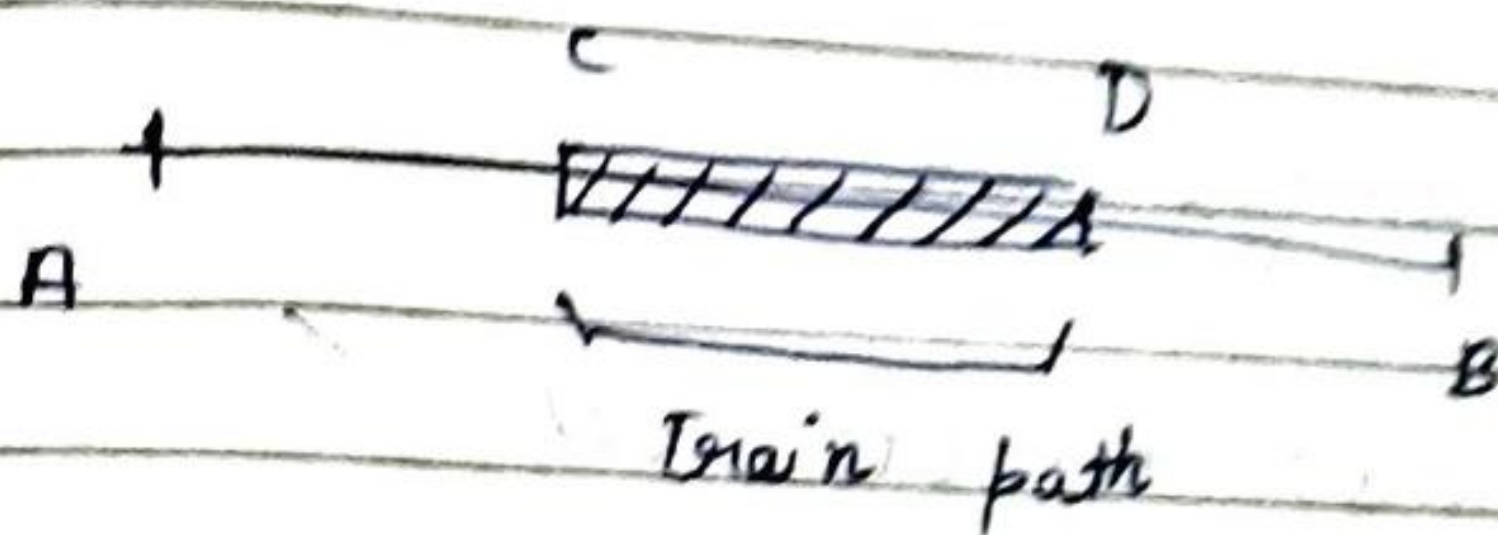
ICPC question
Train or walk

City A $\overset{1 \ 2 \ 3 \ 4 \ 5 \ 6 \ \cdots \ n}{\rule[-0.5ex]{4cm}{0.4pt}}$ City B

$V_{walk} = \left(\frac{1}{p}\right) m/s$

$V_{train} = \left(\frac{1}{q}\right) m/s$

(from C to D)

train starts
at $t = ys$

$\Downarrow$

Train path

Path 1: A to B without train

distance $= |x_A - x_B|$

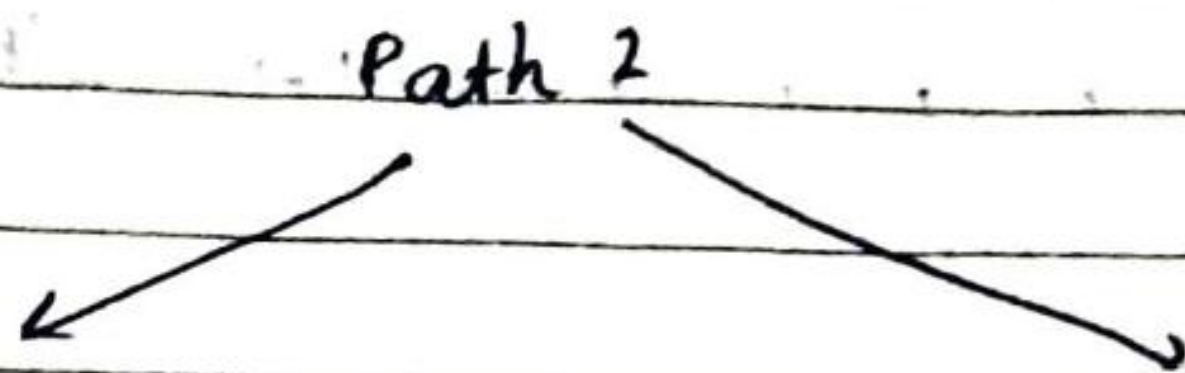$$t = \frac{|x_B - x_A|}{\left(\frac{1}{p}\right)} = p(x_B - x_a)$$

Path 2: A to C and go to D then B.

distance $= |x_C - x_A| + |x_D - x_C| + |x_B - x_D|$

$t = p|x_C - x_A| + q|x_D - x_C| + p|x_B - x_D|$

ans = min (t1, t2);

But train starts at t = y.

Path 2

Train ke liye wait kara

Train chut gayi

if AC (time) > y:
can't catch train

waiting_time

$$= y - p |x_c - x_A|$$

↓

Path 1

t2 += waiting_train

ans = min (t1, t2)

```
cin >> n >> A >> B >> C >> D >> P >> Q >> Y;
int A[n];
for (int i = 0; i < n; i++)  cin >> A[i];
int t1 = p * (abs(A[a] - A[b]));
int tAC = p * (abs(A[c] - A[a]));
if tAC > y {
    // do nothing;        t2 = INT_MAX;
}
else {
    waiting_time = y - tAC;
    t2 += = waiting_time;
}
cout << min(t1, t2) << endl;
```

Modified GCD - Code fories

a = 24, b = 36
gcd = 12
  1   2   3   46 → all divides gcd.

Range - low & high which
         divides both a and b.

Making list of divisors

```
for (i=1; i * i <= n; i++) {
    if (gcd % i == 0) {
        v.pb (i);
        if (i * i ! = gcd) {
            v.pb (gcd/i);
        }
    }

    res = -1;
    for (d: div) {
        if (d >= L && d <= H) {
            res = max (res, d);
        }
    }

    cout << res;
```

Zero remainder array - Codeforces

ex- k = 5

4 4 4 → same group
(same remainder)

map <int, int> m;
m [remainder] = no. of same;

4 4 4   for   k= 5
$x_1 = 1$, $x_2 = 6$, $x_3 = 11$

       5   5

$x_1 \% 5 = x_2 \% 5 = x_3 \% 5$

Claim - Sum of 2 numbers is divisible by k, iff sum of their remainder is divisible by k.

$r_1 = 4 \% 5 = 4$ ⎫ $s = 5 \% 5 = 0$
$r_2 = 1 \% 5 = 1$ ⎭

$r_1 = 4 \% 5 = 4$ ⎫ $s = 5 \% 5 = 0$
$r_2 = 6 \% 5 = 1$ ⎭

           ↓
           ↓

ex- k = 6

8 7 1 8 3 7 5 10 8 4

m [1] = 3   , $x_1 = 5, 11, 17$ ans = 17 + 1
n [2] = 3   , $x_2 = 4, 10, 16$
m [3] = 2   , $x_3 = 3, 9$
m [4] = 1   , $x_4 = 2$
m [5] = 1   , $x_5 = 1$

Chinese Remainder theorem

2 pack of chocolates
1 pack = 5 people (3 rem)
1 pack = 6 people (2 rem)

$x \equiv 3 \pmod 5$
$x \equiv 2 \pmod 6$

$x = n5 + 3 \Rightarrow x = 3, \boxed{8}, 13, 18, 23$
$x = m6 + 2 \Rightarrow x = 2, \boxed{8}, 14, 20, 26$

$x = 8, 38, 68$
$\downarrow$

$\underbrace{8, \quad 38,}_{30} \quad \underbrace{38, \quad 68}_{30}$

$\downarrow ?$

LCM (5,6)

CRT,

$a \equiv a_1 \pmod{p_1}$
$a \equiv a_2 \pmod{p_2}$
$\vdots$
$a \equiv a_k \pmod{p_k}$

$$x = \sum (rem[i] * pp[i] * inv[i]) -f \, prod$$

$$pp[i] = \frac{prod}{num[i]}$$

$$inv[i] = mod \, Inv(pp[i], num[i]);$$

Code

```
for (i=0; i<k; i++){
    int pp= prod/num[i];
    result += rem[i] * inv(pp, num[i])
                                    pp;
}
return result/ prod;
```

$$TC = n \log n$$