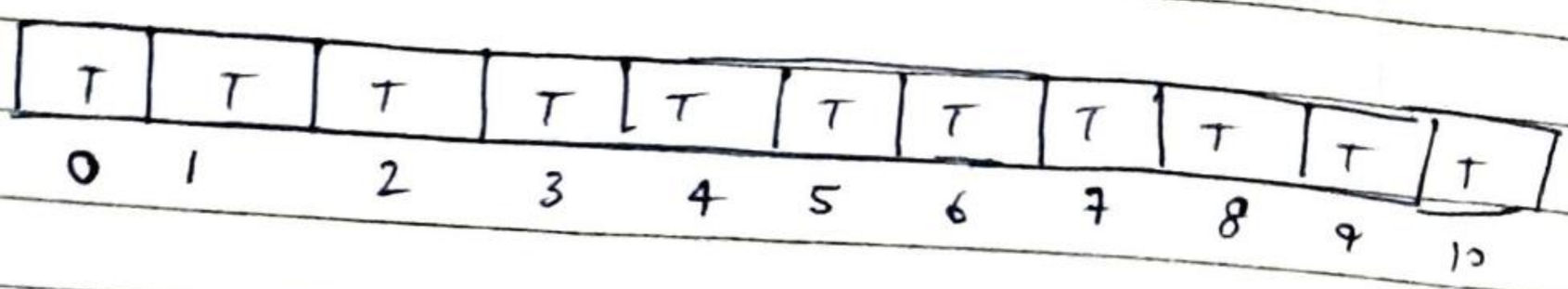
Competitive programming
- Elydark

→ Prime sieve
Sieve of erathestheres

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



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

We need to find all primes from 2 to n.

```
for (i = 2; i * 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 "bob"

gcd (a, 0) = a

→ Extended - euclid algorithm

gcd (a, b) = ax + by

we need to find x & y.

$ax + by = gcd(a,b)$ ——— i)

Also, $gcd(a,b) = gcd(b, a\%b)$ ——— ii)

From i) & ii)

$ax + by = gcd(b, a\%b)$ ——— iii)

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

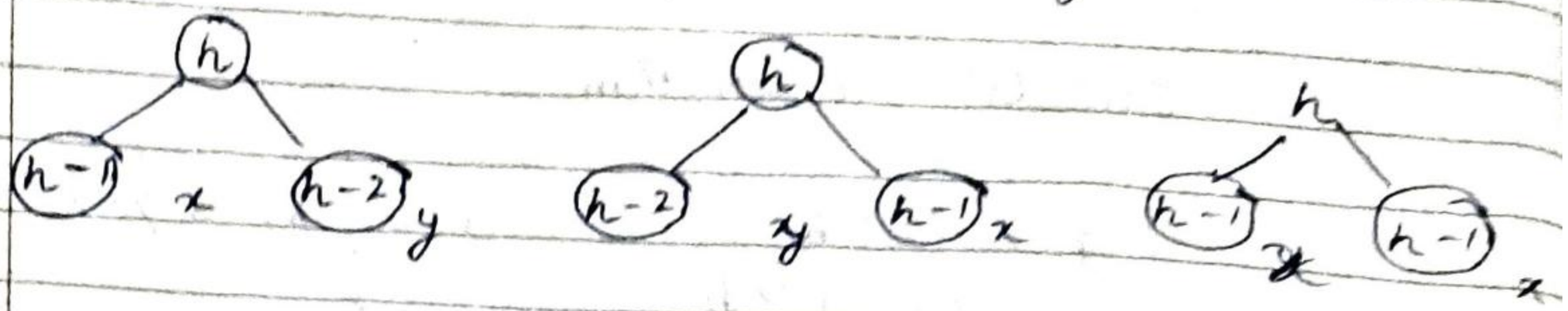
$\Rightarrow bx_1 + \left\{a - \left(\frac{a}{b}\right)\left[\frac{a}{b}\right] \cdot b\right\} y_1 = ax + by$

⇒ On comparing, we get

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

Date | Page No.

→ Number of balanced Binary Tree



ans = 2x + y

```
int balanceBT(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) ⟶ x

(h-2) ⟶ y

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

= $x^2 + 2xy$

modulo inverse

$$a \times b = 1$$

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

Multiplicative mod inverse

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

We have to find B.

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

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

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

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

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

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

$a \cdot b \equiv 1$

$\Rightarrow a \cdot b - 1 \equiv 0$ (multiple of m)

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

$= a \cdot b + mQ = 1$

$= a \cdot b + mQ = 1$

$= gcd(a, m) = 1$

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

Date
Page No

mod inverse (a, m)

= b

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

modinverse (a, m) = b

$\Rightarrow$ (a x ?) % m = 1

i

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 = d - by

This term must be
divisible by x.

$0 \leq y \leq \frac{d}{b}$, $0 \leq x \leq \frac{d}{a}$

$ax = d - by$

↓

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

$d \rightarrow$ fixed

$b \rightarrow$ fixed

$y \rightarrow$ can change.

$ax = d - by_1$

Next term, $d - b(y_1 + a)$

$d - b(y_1 + 2a)$

$d - b(y_1 + 3a)$

⋮

$d - b(y_1 + na)$

On comparing with maximum value of y

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

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

$$n = \frac{\frac{d}{b} - y_1}{a}$$

Total terms $= 1 + n$

Now we need to find y_1

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

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

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

y_1

$ax + by = d$

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

For smallest value,

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

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

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

i) & ii)

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

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

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

This holds

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

$= d^*$ 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) * modInverse(b,a))%a;
long firstvalue = d/b
if (b < y1*b){
    cout<<0<<endl;
}
else{
long n = (firstvalue - y1)/a;
cout<<1+n<<endl;
}
```

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

$$ax + by = d$$

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

$$Ax + By = D$$

Date | Page No.

→ 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) %. 4

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;

}

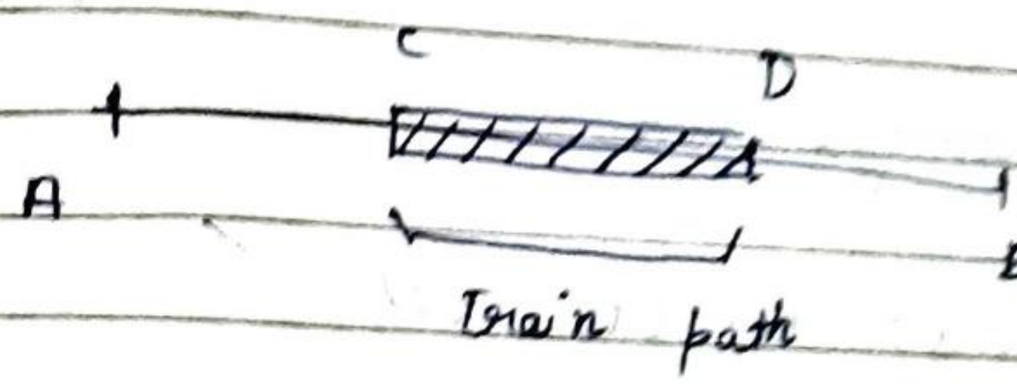
ICPC question
brain or walk

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

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

train stops at 1 to q_i

(From C to D)



Path 1: A to B without train

$$distance = |x_A - x_B|$$

$$s = \frac{D}{t}$$

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

Path 2: A to C and goto 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

waiting_time = y - p|x_c - x_A|

t2 t = waiting_train

Train chut gayi

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

Path 1

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 t = waiting_time;
}
cout << min(t1, t2) << endl;
```

Date
Page No.

Modified GCD - Codeforces

a = 24 , b = 36

gcd = 12

1 2 3 4 6 → 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 - (code for bruces

ex- $K = 5$

4 4 4 → same group (same remainder)

map <int, int > m;

m [remainder] = no. of same;

4 4 4 for $K = 5$

$x_1 = 1, \quad x_2 = 6, \quad x_3 = 11$

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.

$r_1 = 4 \% 5 = 4$
$r_2 = 1 \% 5 = 1$ } $5 = 5 \% 5 = 0$

$r_1 = 4 \% 5 = 4$
$r_2 = 6 \% 5 = 1$ } $5 = 5 \% 5 = 0$

$r_2 =$ ⋮

ex- $K = 6$

8 7 1 8 3 7 5 10 8 4

$m[1] = 3, \quad x_1 = 5, 11, 17$

$m[2] = 3, \quad x_2 = 4, 10, 16$

$m[3] = 2, \quad x_3 = 3, 9$

$m[1] = 1, \quad x_4 = 2$

$m[5] = 1, \quad x_5 = 1$

ans = 17 + 1

Chinese Remainder theorem

2 pack of chocolates

1 pack = 5 people (3 rem)

1 pack = 6 people (2 rem)

$x \equiv 3 \pmod 5$

$x \equiv 2 \pmod 6$

$x = 5n + 3 \Rightarrow x = 3, 8, 13, 18, 23$

$x = 6m + 2 \Rightarrow x = 2, 8, 14, 20, 26$

$x = 8, 38, 68$

8, 38, 68

30 30

?

LCM (5, 6)

CRT,

$a \equiv a_1 \pmod{p_1}$

$a \equiv a_2 \pmod{p_2}$

$a \equiv a_k \pmod{p_k}$

$x = \sum (rem[i] * pp[i] * inv[i]) \% prod$

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

$inv[i] = modInv(pp[i], num[i]);$

Code

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

TC = $n \log n$

Date
Page No.

Train partner - Codechef

1	4	9	12
2	5	10	13
3	6	11	14
7	8	15	16

Remainder (%8)	Solution	Coach
0	$n-1$	SL
7	$n+1$	SU
1	$n+3$	LB
4	$n-3$	LB
2	$n+3$	MB
5	$n-3$	MB
3	$n+3$	UB
6	$n-3$	UB

Date
Page No.

Matrix exponentiation

$$A = \begin{bmatrix} & \\ & \end{bmatrix} \qquad B = \begin{bmatrix} \\ \\ \end{bmatrix}$$

$$A \cdot B = \sum_{x=1}^{k} A_{ix} \times B_{xj}$$

We know, $f(n) = f(n-1) + f(n-2)$

$$M \times \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n+1) \\ f(n) \end{bmatrix}$$

We need to find this M

$M = 2 \times 2$

$$\Rightarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n+1) \\ f(n) \end{bmatrix}$$

$$\neq \quad a * f(n) + b * f(n-1) = f(n+1)$$

Clearly, $a = 1, \quad b = 1$

$$c * f(n) + d * f(n-1) = f(n)$$

Clearly, $c = 1, \quad d = 0$

$$M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

So, $$M * \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n+1) \\ f(n) \end{bmatrix}$$

$$M^n * \begin{bmatrix} f(n+1) \\ f(n) \end{bmatrix} = \begin{bmatrix} f(n+2) \\ f(n+1) \end{bmatrix}$$

$$M^{*} \quad M^{*} \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n+1) \\ f(n) \end{bmatrix}$$

$$\vdots$$

$$M^{k} * \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} f(n+k) \\ f(n+k-1) \end{bmatrix}$$

Put $n = 1$

& $k = n-1$

$$M^{n-1} * \begin{bmatrix} f(1) \\ f(0) \end{bmatrix} = \begin{bmatrix} f(1+n-1) \\ f(1+n-1-1) \end{bmatrix}$$

$$M^{n-1} * \begin{bmatrix} f(1) \\ f(0) \end{bmatrix} = \begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix}$$

This is matrix exponentiation

$$M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$M^{n-1} \rightarrow \log n$

$$A^{n} = \begin{cases} A^{n/2} \cdot A^{n/2} & n = \text{even} \\ \left[A^{n/2} \cdot A^{n/2}\right] \cdot A & n = \text{odd} \end{cases}$$

Date
Page No.

Fibo sum

n & m

$S(i) = S_{i-1} + f(n)$

$f(n) = f(n-1) + f(n-2)$

$f(n-1) = f(n-2) + f(n-3)$

$f(1) = 1$

(+)

$f(n) = f(n-2) + f(n-3) + f(n-4)$

$f(n) = S(n-1) + 1$

$S(n-1) = f(n) - 1$

$S(n) = f(n+2) - 1$

According to question,

$S(m) - S(n-1)$

$= f(m+2) - 1 - f(n+1) + 1$

$ans = [f(m+2) - f(n+1)] \% m$

Date
Page No.

Game theory

(2 player game)

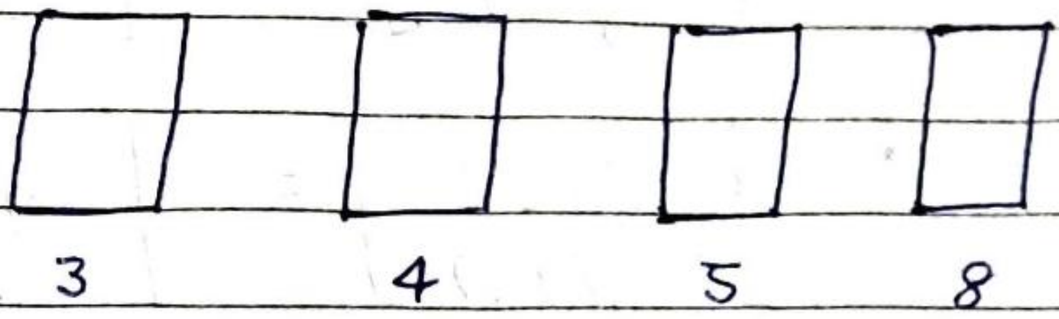
Combinatorial game theory - Finite game

Impartial game (No - limitation)

Partison Game (Limitation) ex- chess

Game theory deals with this part mostly.

→ Game of Nim

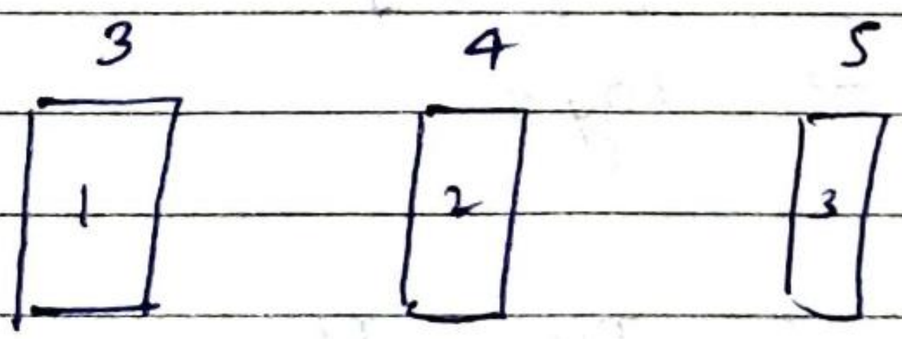


Player 1 Player 2

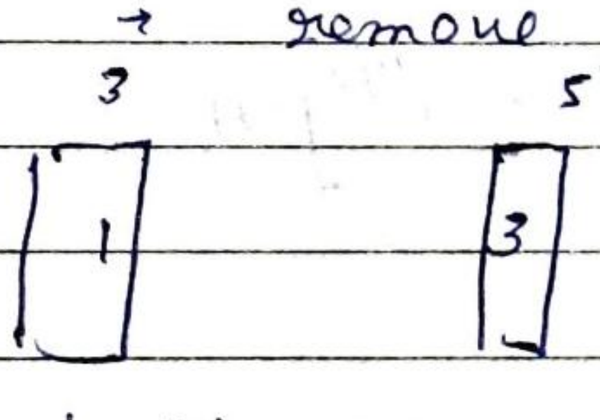
Can remove any no. of file.

In the end who will be the winner

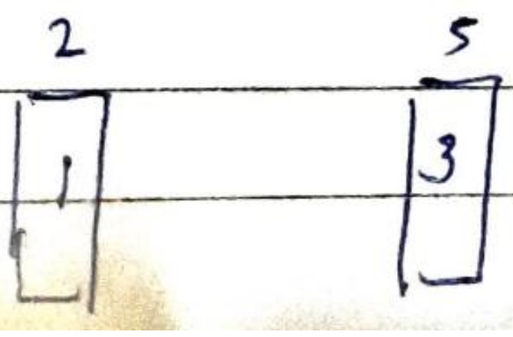
ex-



P1 - 2 → remove all



P2 → 1 → remove 1



Date
Page No

P1 → 1 → remove all

5

3

P2 → 3 → remove all

Player 1 lost

Player 2 wins

Will player 1 always loose?

Not neccessary.

→ Nim sum theorem

Find Nim sum

i.e. Commulative XOR of all piles

3 ^ 4 ^ 5.

$$\text{Nim sum} = \begin{cases} 0, & \text{Player 1 looses} \\ \neq 0 & \text{Player 1 wins} \end{cases}$$

On condition that both plays optimally and if $n \rightarrow$ even, play 1 wins

Proof of Nim sum.

A → xor → B

Not needed

Mex = minimum excluded set

max {0, 1, 3}, ans = 2
max {0, 1} ans = 2

Grundy numbers.

grundy numbers = max {grundy (n-1), grundy (n-2)}

grundy (0) = 0.
grundy (1) = {g(0)} = {0} = 1
grundy (2) = {g(0), g(1)}
= {0, 1} = 2
⋮

grundy (n) = {g(0), g(1), g(2) --- g(n-1)} = n

Grundy number is a no. that defines a state of a game.

n stones → 1, 2, 3 can be removed.

grundy (n) = max {g(n-1), g(n-2), g(n-3)}

Here, g(0) = 0, g(1) = 1, g(2) = 2
g(3) = 3, g(4) = 0
g(5) = 1, g(6) = 2, g(7) = 3

Date | Page No.

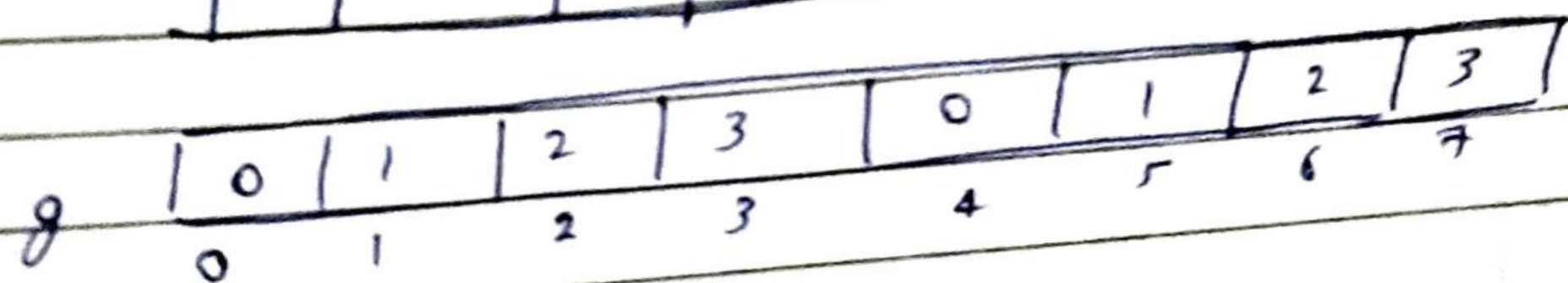
Sprague grundy theorem

Cumulative XOR = $(g(p_1) \wedge g(p_2) \wedge \ldots)$

$$XOR = \begin{cases} 0, & \text{player 1 loose} \\ \neq 0 & \text{player 1 wins} \end{cases}$$

on condition that both players plays optimally.

Ex -

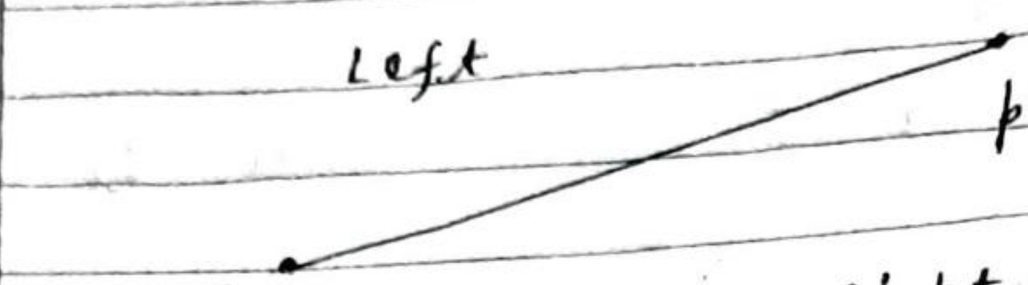


XOR = $g(3) \wedge g(4) \wedge g(5)$

= $3 \wedge 0 \wedge 1$

Computational geometry

→ Point location test (CSES)

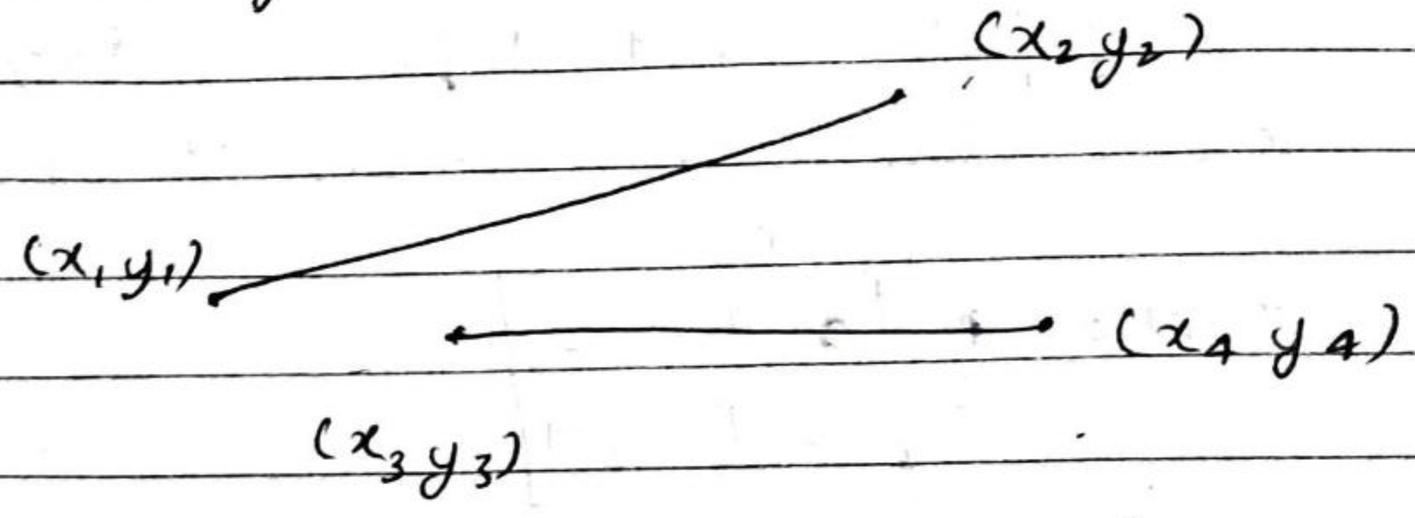


$$m = \frac{dy}{dx} = \frac{y_2 - y_1}{x_2 - x_1}$$

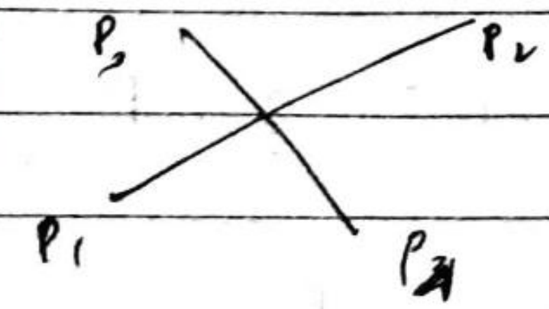
$$m' = \frac{dy'}{dx'} = \frac{y_3 - y_2}{x_3 - x_2}$$

```
if (m' - m > 0) { cout << "Left" ; }
else if (m' - m < 0) { cout << "Right" ; }
else cout << " On the line" ;
```

→ Line segment intersection - (CSES)



Algo



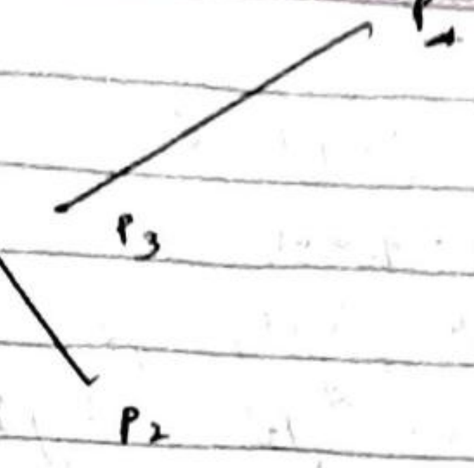
$\overline{P_1 P_2}\ P_3$ = Left

$\overline{P_1 P_2}\ P$ Orientation must be different for each point

Date

Page No.

ex-

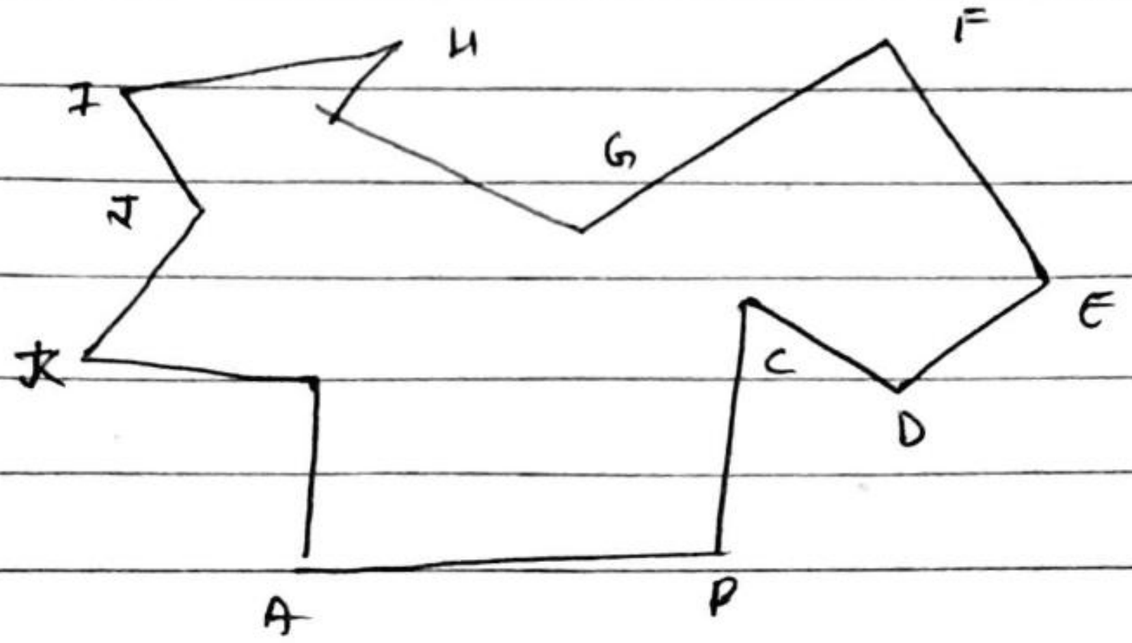


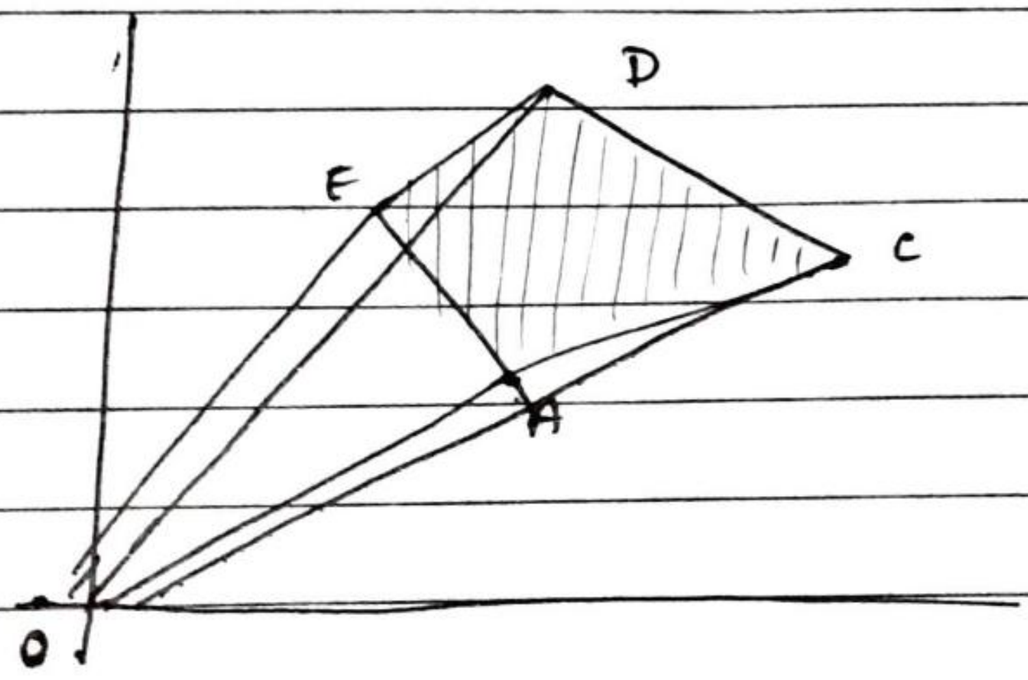
$m_1 = \frac{y_2 - y_1}{x_2 - y_1}$, $m_2 = \frac{y_4 - y_2}{x_4 - x_3}$

if ($m_1 == m_2$) { return true; }
if (01 != 02 && 03 != 04) { return true; }
return false;

→ Polygon area - CSES

cin >> vertex;



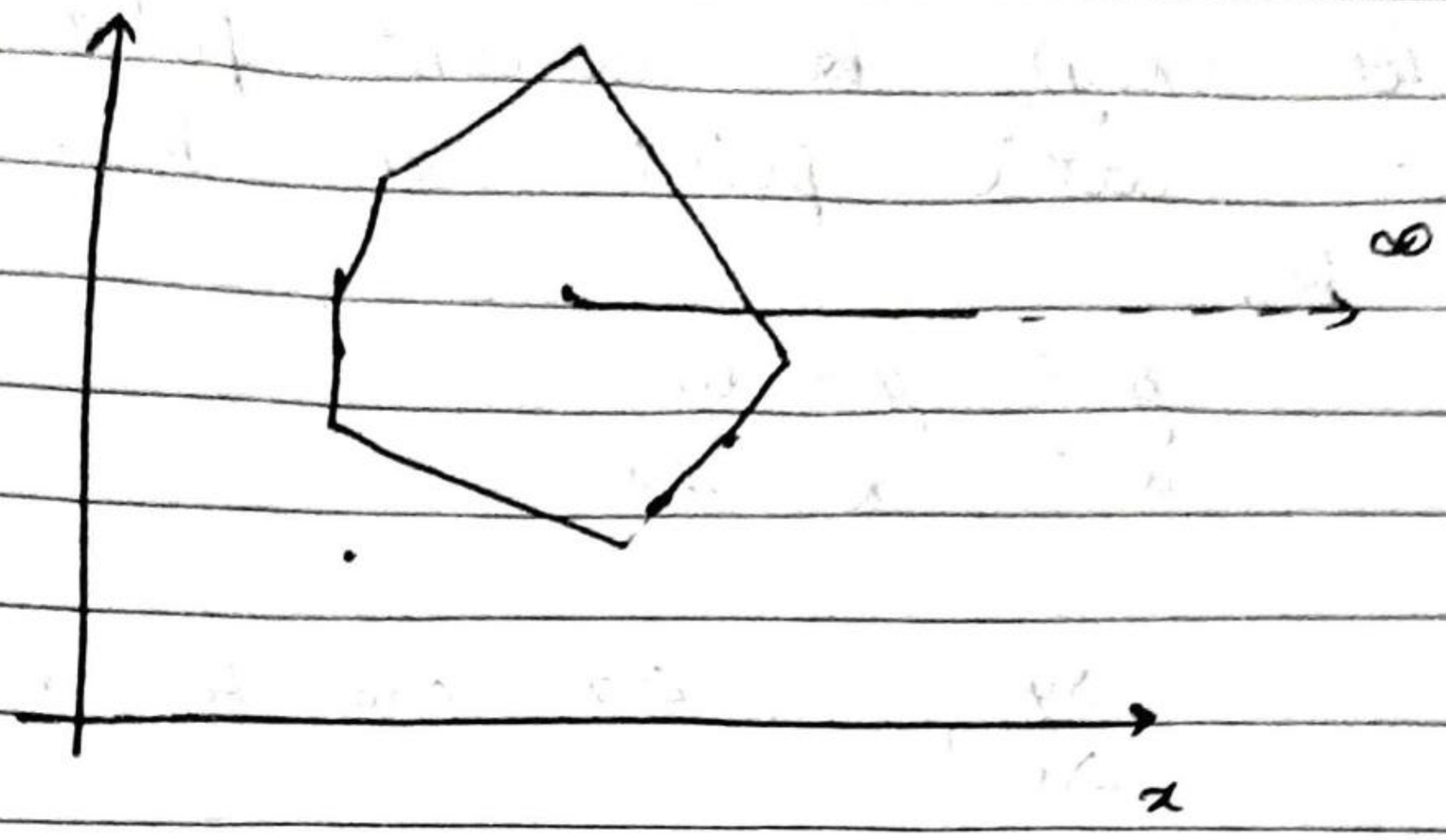


OAC + OCD
- ODE - OEA

(Direction is important in cross product)

```
float ans = 0;
for (i=0; i<n; i++) {
    x1 = point[i].first, y1 = point[i].second
    x2 = point[(i+1)%n].first, y2 = points[(i+1)%n].sec
    ans += ((x1 * y2) - (y1 * x2));
}
cout << abs(ans)/2 << endl;
```

Point in a polygon



No. of intersection = odd
(Inside polygon)

= even
(outside polygon)

For ~~outside~~ on the line,

(x, ∞)

Lattice point

Pick's theorem,

$$A = i + \frac{b}{2} - 1$$

A ← area

i ← Lattice points (inside the polygon)

b ← LP on boundary

Boundary lattice point

$$m = \frac{dy}{dx}$$

let (x, y) be current point then next lattic point on boundary will be

$$y' = y + dy$$
$$x' = x + dx$$

But $\frac{\Delta y}{\Delta x}$, Δx can be 0.

then $a = (x_2 - x_1)$
$b = (y_2 - y_1)$

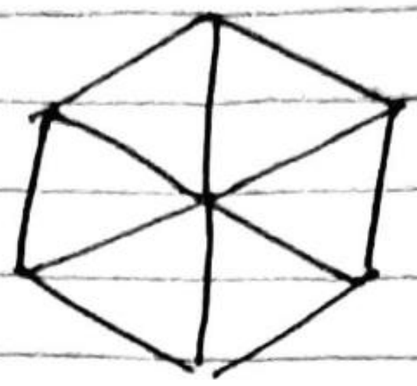
$gcd(a, b) + 1$

(But it will include the given points as well).

sure $gcd(a, b) - 1$ to find only boundary lattic excluding given points

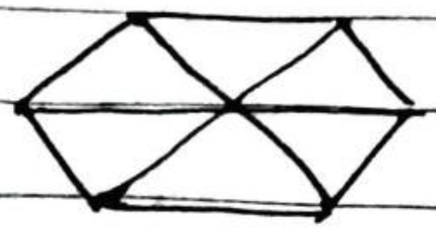
Date
Page No

Gerald hexagon- Codeforce.

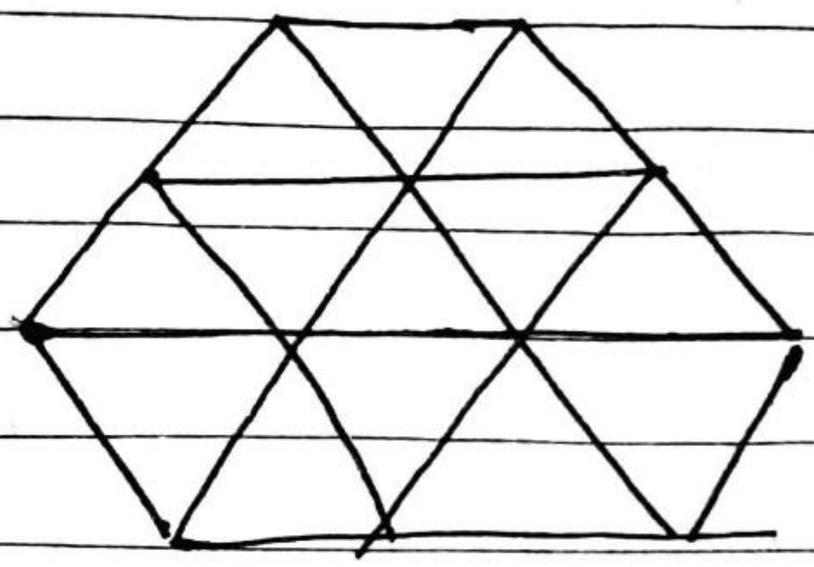


Gerald hexagon- Code forces

ex-

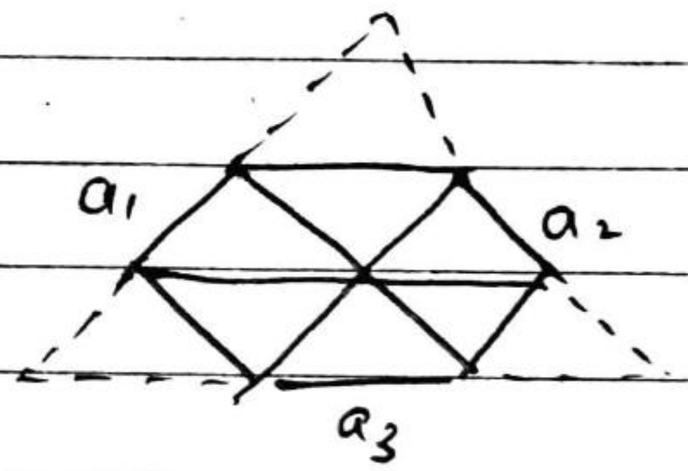


$n = 6$



$n = 13$

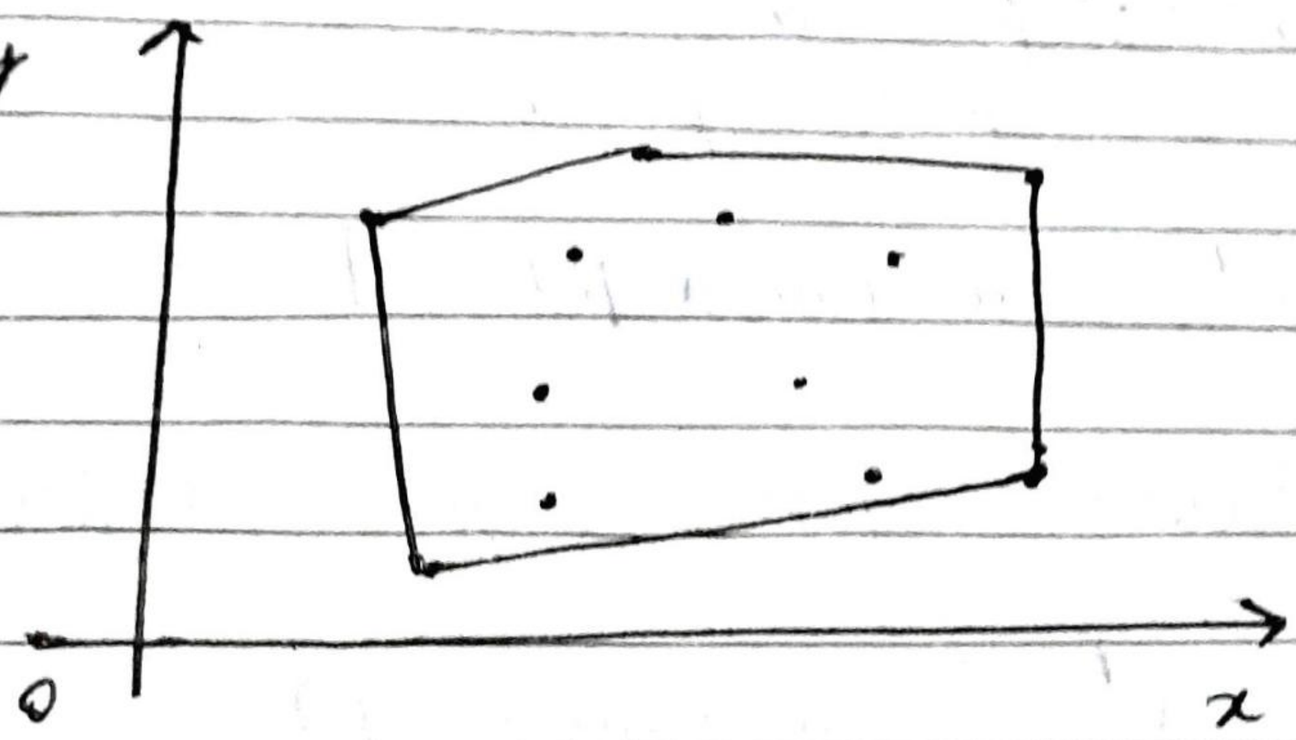
No. of triangles = $\frac{\text{Area of hexagon}}{\text{area of each trior}}$



$$= \frac{\frac{\sqrt{3}}{4}(a_1+a_2+a_3)^2 - \frac{\sqrt{3}}{4}(a_1^2+a_2^2 \cdot}{\frac{\sqrt{3}}{4}(1)^2}$$

Date
Page No.

Convex hull. (Gift wrapping)
(Jarvis algorithm)



1) Take the leftmost point as starting
2) now using this, find the left
points and keep joining them

Date
Page No.

Fermat's little theorem

$(a^p) \% p = a$

p = prime

$$\boxed{a^p \equiv a \pmod{p}}$$

this holds for any integer 'a'.

$$a^p \equiv a \pmod{p}$$

dividing by a both side

$$a^{p-1} \equiv 1 \pmod{p}$$

$$\Rightarrow \boxed{(a^{p-1}) \% p = 1}$$

$$= a^{-1} * a^{(p-1)} \bmod p = a^{-1}$$

$$= (a^{-1} * a^{p-1}) \bmod p = a^{-1} \pmod{p}$$

$$(a^{p-2}) \bmod p =$$

So, $(a^{-1}) \% m = (a^{m-2}) \bmod m$

Find this using exponential

Page No.

→ Code forces - ${}^{n}C_{r} \% m$ in $O(\log n)$

$${}^{n}C_{r} \% m = \left(\frac{\lfloor n}{\lfloor r \; \lfloor n-r}\right) \% m$$

$$= \left(\lfloor n \% m * (\lfloor r)^{-1} \% m * (\lfloor n-r)^{-1} \% m\right) \% m$$

$$= \lfloor n \% m * (\lfloor r)^{m-2} \% m * [(\lfloor n-r)^{m-2} \% m]] \% m$$

↓

Calculating this is not a tough task

Application of Fermat's theorem

$$(a/b) \% m = (a * b^{-1}) \% m$$

$$= ((a \% m) \; (b^{-1} \% m)) \% m$$

$$b^{-1} \% m = (b^{m-2}) \% m$$

$$= ((a \% m) \; (b^{m-2}) \% m) \% m$$