

## Chinese Remainder Theorem

→ To know this need idea of

- ① Inverse
- ② Congruency

### ① Inverse

$$3 \text{ modulo } 7 = ?$$

$$\therefore \gcd(3, 7) = ?$$

Find gcd using extended eucli

$$7 = 2 \times 3 + 1 \quad \textcircled{1}$$

$$\therefore 1 = \frac{7 - 2 \times 3}{x}$$

$$\therefore 3 \text{ modulo } 7 = -2$$

$$\frac{1}{7} = \frac{7}{7} - \frac{2 \times 3}{7}$$

$$\frac{1}{7} = -2 \times \frac{3}{7}$$

$$\therefore \frac{1}{3} = -2$$

$$\therefore 3^{-1} = -2$$

$$21 \text{ modulo } 46$$

$$\therefore \gcd(21, 46) = ?$$

$$46 = 2 \times 21 + 4 \quad \textcircled{1}$$

$$21 = 5 \times 4 + 1 \quad \textcircled{11}$$

$$\Rightarrow 1 = 21 - 5 \times 4$$

$$\textcircled{11} \rightarrow$$

$$4 = 46 - 2 \times 21$$

$$\therefore 1 = 21 - 5 \times (46 - 2 \times 21)$$

$$= 21 - 5 \times 46 + 10 \times 21$$

$$= 11 \times 21 - \frac{5 \times 46}{x}$$

$$\therefore 21 \text{ modulo } 46 \\ = 11$$

## modular multiplicative inverse

$$5 \times 5^{-1} = 1$$
$$5 \times \frac{1}{5} = 1$$
$$A \times \frac{1}{A} = 1$$

$\frac{1}{5}$  is the multiplicative inverse of 5

Under mod n

$$A \times A^{-1} \equiv 1 \pmod{n}$$

$$(A \times A^{-1})/n = 1$$

$$\therefore 3 \times \frac{?}{p} \equiv 1 \pmod{5}$$

Hence p is the modulo multiplicative Inverse of 3 modulo 5

$$3 \times 2 \equiv 1 \pmod{5} \quad (3 \times 2 / 5 = 1)$$

multiplicative Inverse

$$2 \times ? \equiv 1 \pmod{11} \rightarrow ? = 6$$

$$Ax = 1 \pmod{m}$$
$$x = \frac{1}{A} \pmod{m}$$
$$x = A^{-1} \pmod{m}$$
$$x \text{ is M1}$$
$$A \text{ non modulo } m$$

a modulo b  
 $\therefore$  there exist a multiplicative Inverse if a, b relatively prime or coprime

$$\gcd(a, b) = 1$$

$$\gcd(2, 11) = 1$$

$$\begin{array}{l} \cancel{3 \text{ modulo } 5} \\ \cancel{\text{Gcd}(3, 5) = ?} \\ \therefore \cancel{5 = 3 \times 1 + \cancel{2}} \end{array}$$

$$3 = 2 \times 1 + 1$$

$$\begin{array}{l} 3 \text{ modulo } 5 \\ \text{Gcd}(3, 5) = ? \end{array}$$

$$\begin{array}{l} 5 = 3 \times 1 + 2 \\ 3 = 2 \times 1 + \cancel{2} \end{array}$$

$$3 = 1 \times (5 - 3) + 1$$

$$3 = 5 - 3 + 1$$

$$\begin{array}{l} 1 = -5 + 6 \\ = -5 + (2 \times 3) \end{array}$$

$$\therefore \text{MI} = 2$$

$$43 = \cancel{17} \times 2 + \cancel{9} \rightarrow$$

$$9 = 43 - 17 \times 2$$

$$17 = \cancel{9} \times 1 + \cancel{8} \rightarrow$$

$$8 = 17 - 9$$

$$9 = 8 \times 1 + 1$$

$$= 17 - (43 - 17 \times 2)$$

$$\Rightarrow 1 = 9 - 8$$

$$\begin{array}{l} \cancel{9 -} \\ \cancel{17 - 9} \end{array}$$

$$= 43 - 17 \times 2 - 17 + 9 - 43 + 17 \times 2$$

$$\begin{array}{l} \cancel{9 -} \\ \cancel{17 + 9} \end{array}$$

$$\cancel{43 - 3 \times 17 + 3}$$

$$= 43 \times 2 - 5 \times 17$$

$$\therefore x = -5$$

## Congruency

(W)

a, b

$$a \bmod m = \pi_1$$

$$b \bmod m = \pi_2$$

: if  $\pi_1 = \pi_2$  : a is congruent of  $b \bmod m$

$$a \equiv b \pmod{m}$$

Theorem - 1

$$(a \bmod m) + p \in A$$

Theorem - 2

$$a = b + km$$

$$a - b = km$$

$$R = \frac{a-b}{m}$$

: if  $(a-b) \bmod m = 0$  then

$$a \equiv b \pmod{m}$$

## CRT    (weak)

→ is used to solve a set of different congruent equations with one variable but different moduli which are relatively prime (coprime)

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

Find x

:

$$x \equiv a_n \pmod{m_n}$$

→ CRT states that above equations have a unique solution if the moduli are relatively prime

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

$$x \equiv a_3 \pmod{m_3}$$

$$x = ?$$

if  $x$  exist if

$$\gcd(m_1, m_2) = \gcd(m_2, m_3) = 1$$

or coprime

Test #AS

$$m = m_1 \cdot m_2 \cdot m_3 = 3 \times 5 \times 7 = 105$$

$$M_1 = \frac{105}{3} = 35$$

$$Y_1 = 35 \pmod{3}$$

$$= M_1 \pmod{m_1}$$

$$3 =$$

$$35 > 3$$

can't do

35 modulo 3

$$\begin{aligned} 35 &= 3 \times 11 + 2 \\ 3 &= 2 \times 1 + 1 \\ 1 &= 3 - 2 \\ &= 3 - 35 + 3 \times 11 \\ &= 12 \end{aligned}$$

ans :- 12

$$X = \left( a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} + a_3 M_3 M_3^{-1} \right) \pmod{m}$$

∴ Hence,

$$M = m_1 \times m_2 \times m_3$$

$$= 3 \times 5 \times 7$$

$$= 105$$

$$\begin{aligned} m_1 &= \frac{M}{m_1} \\ &= \frac{105}{3} \\ &= 35 \end{aligned}$$

$$\begin{aligned} m_2 &= \frac{M}{m_2} \\ &= \frac{105}{5} \\ &= 21 \end{aligned}$$

$$\begin{aligned} m_3 &= \frac{M}{m_3} \\ &= 105/7 \\ &= 15 \end{aligned}$$

$$m_1 \times m_1^{-1} = 1 \pmod{m_1}$$

$$35 \times m_1^{-1} = 1 \pmod{3}$$

$$\begin{aligned} 35 \times 2 &= 70 \cdot 3 = 1 \\ \therefore m_1^{-1} &= 35 \text{ modulo } 3 \end{aligned}$$

$$m_1^{-1} = 2$$

$$m_2 \times m_2^{-1} = 1 \pmod{m_2}$$

$$\begin{aligned} 21 \times m_2^{-1} \\ = 1 \pmod{5} \end{aligned}$$

$$\begin{aligned} 21 \times 1 &= 21/5 = 1 \\ m_2^{-1} &= 1 \end{aligned}$$

$$m_3 \times m_3^{-1} = 1 \pmod{m_3}$$

$$15 \times m_3^{-1} = 1 \pmod{7}$$

$$15 \times 1 = 15 \cdot 1 \cdot t = 1$$

$$m_3^{-1} = 1$$

## Extended Euclid's

$ax + by = 1$

Given  $AX + BY = \text{GCD}(A, B)$

Find  $(x, y)$

$$AX + BY = \text{GCD}(B, A \% B)$$

$$\begin{aligned} x &= y_1 \\ \therefore y &= x_1 - \left\lfloor \frac{a}{b} \right\rfloor y_1 \end{aligned}$$

Ans

(2, -1)

(-1, 2)

(1, -1)

$$x_3 = 1 \\ x_2 = x_3 - \lfloor \frac{a}{b} \rfloor y_3 \\ = 0 - \lfloor \frac{18}{12} \rfloor 1 \\ = -1$$
$$\Rightarrow 18x_2 + 12y_2 = \text{GCD}(18, 12) \\ = \text{GCD}(18, 12)$$

(0, 1)

$$x_3 = y_4 = 0 \\ y_3 = x_4 - \lfloor \frac{a}{b} \rfloor y_4 \\ = 1 - \lfloor \frac{12}{6} \rfloor 0 \\ = 1$$
$$(1, 0) \Rightarrow 6x_1 + 0y_1 = \text{GCD}(6, 0)$$

$$6x_1 = 6$$

$$\therefore x_1 = 1$$

$$y_1 = 0$$

Base

Index ;	Quotient $q_i$	Reminder $r_i$	$x_i = x_{i-2} - q_i x_{i-1}$	$y_i = y_{i-2} - q_i y_{i-1}$
0	<del>240</del>	240	$1 - 2$	0
1	<del>48</del>	46	$0 - 1$	1
2	$240/46 = 5$	$240 \cdot 46 = 10$ ( $240 - 5 \times 46$ )	$x_i = 1 - 5 \times 0$ = 1	$y_i = 0 - 5 \times 1$ = -5
3	$46/10 = 4$	$46 \cdot 10 = 6$	$0 - 4 \times 1$ = -4	$1 - 4 \times 0 - 5$ = <del>1</del> 21
4	$10/6 = 1$	$10 \cdot 6 = 4$	$1 + 4 = 5$	$-5 - 21 = -26$
5	$6/4 = 1$	$6 \cdot 4 = 2$	$-4 - 5 = -9$	$21 + 26$ = 47
6	$4/2 = 2$	$4 \cdot 2 = 0$	$5 + 18$ <del>= 23</del>	$-26 - 2 \times 97$ = 120

int ext\_gcd ( int A, int B, int &x, int &y )

{

int  $x_1 = 0, y_1 = 1;$

int  $x_2 = 1, y_2 = 0;$

$$x_i = x_{i-2} - q_i \frac{x_{i-1}}{x_i}$$

int  $x, y, n_2, n_1, q, r;$

for (  $n_2 = A, n_1 = B; n_2 \neq 0;$     $n_2 = n_1, n_1 = r,$     $x_2 = x_1, x_1 = x,$     $y_2 = y_1, y_1 = y,$    )

{

$q = n_2 / n_1;$

$r = n_2 \% n_1;$

$x = x_2 - (q * x_1);$

$y = y_2 - (q * y_1);$

}

$x = x_2;$

$y = y_2;$

return  $n_2;$

modular multiplicative  
Inverse using Ext euclid

$$Ax \equiv 1 \pmod{3}$$

$$Ax + my = 1$$

$\times$  modulo Inverse  
of A modulo m

m1 of 3 mod 5

A	B	Remainder r	Quotient Q	T <sub>1</sub>	T <sub>2</sub>	T $T = T_1 - T_2 \times Q$
5	3	$5 \cdot 3 = 2$	$5 \cdot 3 = 1$	0	1	-1
3	2	1	1	1	-1	2
2	1	0	2	-1	2	-5
1	0	x	x	(2)	-5	<del>any</del>

## Linear Diophantine Equation

$$4x + 10y = 8$$

$\therefore \text{GCD} = 2 = g$  ; since  $2$  divides  $8$   
 $(4, 10)$  solution exists //

$$\frac{a}{g} = \frac{b}{g} = \frac{c}{g}; \quad \frac{4}{2} = \frac{10}{2} = \frac{8}{2} \rightarrow 4$$
$$\therefore 2x + 5y = 1 \rightarrow \textcircled{1} \quad \text{Find solution using ext-gcd}$$

$$\therefore \text{From ext-gcd} \rightarrow x, y = -2, 1$$

this solution is for  $ax + by = 1$

$\therefore$  we need to multiply  $4$

$$\boxed{(-8, 4)}$$

Ans

bool linearDiophantine  
(int A, int B, int c, int x, int y)

int g = gcd(A, B)

if (c % g != 0) return false

$$\text{int } a = A/g \quad b = B/g \quad c = \frac{c}{g}$$

ext\_gcd(a, b, x, y)

: if (g < 0)

$$\{ \quad a = -x - 1 \quad b = b \times -1 \quad c = c \times -1 \}$$

$$x = x * c$$

$$y = y * c$$

return true

}

## Sieve of Eratosthenes

```

int max = 1000000;
bool isPrime[max];
vector<int> prime;

```

```

void sieve();
isPrime[i]

```

Pseudocode

Void sieve()

{

int n;

vector<bool> isPrime(n+1, true);

Initial all  
number true  
on prime

isPrime[0] = isPrime[1] = false;

for (int i = 2 ;  $i \leq n$ ; i++) {

if (isPrime[i] == true)

{ for (int j = i\*i; j <= n; j += i)

{ isPrime[j] = false }

}

}

$$n = 16$$

$\rightarrow 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16$

$i = 2$  prime and all its sqrt not prime

$\rightarrow ② \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16$

$i = 3$  prime

$\rightarrow 2 \ ③ \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16$

$i = 4$

$\rightarrow \text{isprime}[4] = 4 = \text{not prime}$

= 2nd loop not work

$i = 5$

$\rightarrow 5 \times 5 = 25 > n$  loops break

$② \ ③ \ \cancel{4} \ \cancel{5} \ 6 \ \cancel{7} \ \cancel{8} \ \cancel{9} \ \cancel{10} \ \cancel{11} \ \cancel{12} \ \cancel{13} \ \cancel{14} \ \cancel{15} \ \cancel{16}$

Primes  $\rightarrow$

(2, 3, 5, 7, 11, 13)

## Bit sieve

```
int N = 100;
```

```
int status[100/32];
```

```
bool check(int index,  
           int position)
```

```
{
```

```
    return
```

```
(bool)(index & (1<<position));
```

```
y
```

*int*

```
bool set(int index, int position)
```

```
{ return index = index | (1<<position); }
```

```
y
```

```
void sieve
```

{ ~~\* we only hunt for odd number  
even num by default not prime~~ }

```
for (int i = 3 ; i <= sqrt(sqrt(N)) ; i += 2 )
```

→ initial all value 0

→ status arr is integer type  
Hence it has 32bit in  
1 index

→ all these 32 bit indicates  
32 different numbers  
whether they prime or not

→ status [0] = 0 - 31 number

→ a number index = i / 32;

→ number status of prime  
is at status [i/32] in  
i/32 bit

5 25 49 81

121 169 225

289 361 441

529 625 729

841 961 1089

1225 1369 1521

1681 1849 2025

2209 2341 2481

2641 2809 2969

3121 3361 3601

3841 4161 4481

4801 5121 5441

6161 6561 6961

7361 7761 8161

8561 8961 9361

9761 10161 10561

11361 11761 12161

12961 13361 13761

14561 14961 15361

16161 16561 16961

18161 18561 18961

19761 20161 20561

21361 21761 22161

22961 23361 23761

24561 24961 25361

26161 26561 26961

27761 28161 28561

29761 30161 30561

31361 31761 32161

32961 33361 33761

34561 34961 35361

36161 36561 36961

38161 38561 38961

39761 40161 40561

41361 41761 42161

42961 43361 43761

44561 44961 45361

46161 46561 46961

47761 48161 48561

49761 50161 50561

51361 51761 52161

52961 53361 53761

54561 54961 55361

56161 56561 56961

57761 58161 58561

59761 60161 60561

61361 61761 62161

62961 63361 63761

64561 64961 65361

66161 66561 66961

67761 68161 68561

69761 70161 70561

71361 71761 72161

72961 73361 73761

74561 74961 75361

76161 76561 76961

77761 78161 78561

79761 80161 80561

81361 81761 82161

82961 83361 83761

84561 84961 85361

86161 86561 86961

87761 88161 88561

89761 90161 90561

91361 91761 92161

92961 93361 93761

94561 94961 95361

96161 96561 96961

97761 98161 98561

99761 100161 100561

101361 101761 102161

102961 103361 103761

104561 104961 105361

106161 106561 106961

107761 108161 108561

109761 110161 110561

111361 111761 112161

112961 113361 113761

114561 114961 115361

116161 116561 116961

117761 118161 118561

119761 120161 120561

121361 121761 122161

122961 123361 123761

124561 124961 125361

126161 126561 126961

127761 128161 128561

129761 130161 130561

131361 131761 132161

132961 133361 133761

134561 134961 135361

136161 136561 136961

137761 138161 138561

139761 140161 140561

141361 141761 142161

142961 143361 143761

144561 144961 145361

146161 146561 146961

147761 148161 148561

149761 150161 150561

151361 151761 152161

152961 153361 153761

154561 154961 155361

156161 156561 156961

157761 158161 158561

159761 160161 160561

161361 161761 162161

162961 163361 163761

164561 164961 165361

166161 166561 166961

167761 168161 168561

169761 170161 170561

171361 171761 172161

172961 173361 173761

174561 174961 175361

176161 176561 176961

177761 178161 178561

179761 180161 180561

181361 181761 182161

182961 183361 183761

184561 184961 185361

186161 186561 186961

187761 188161 188561

189761 190161 190561

191361 191761 192161

192961 193361 193761

194561 194961 195361

196161 196561 196961

197761 198161 198561

199761 200161 200561

201361 201761 202161

202961 203361 203761

204561 204961 205361

206161 206561 206961

207761 208161 208561

209761 210161 210561

211361 211761 212161

212961 213361 213761

214561 214961 215361

216161 216561 216961

217761 218161 218561

219761 220161 220561

221361 221761 222161

222961 223361 223761

224561 224961 225361

226161 226561 226961

227761 228161 228561

229761 230161 230561

231361 231761 232161

232961 233361 233761

234561 234961 235361

236161 236561 236961

237761 238161 238561

239761 240161 240561

241361 241761 242161

242961 243361 243761

244561 244961 245361

246161 246561 246961

247761 248161 248561

249761 250161 250561

251361 251761 252161

252961 253361 253761

254561 254961 255361

256161 256561 256961

257761 258161 258561

259761 260161 260561

261361 261761 262161

262961 263361 263761

264561 264961 265361

266161 266561 266961

267761 268161 268561

269761 270161 270561

271361 271761 272161

272961 273361 273761

274561 274961 275361

276161 276561 276961

277761 278161 278561

279761 280161 280561

281361 281761 282161

282961 283361 283761

284561 284961 285361

286161 286561 286961

287761 288161 288561

289761 290161 290561

291361 291761 292161

292961 293361 293761

294561 294961 295361

296161 296561 296961

297761 298161 298561

299761 300161 300561

301361 301761 302161

302961 303361 303761

304561 304961 305361

306161 3065

```
{  
    if (check(status[i/32], i%32) == 0)  
        for (int j = i*i ; j <= n ; j += (2i))  
    {  
        status[j/32] = set(status[j/32], j%32)  
    }  
}
```

```
prime.pushback(2);  
for (int i=3 ; i <= N ; i += 2)  
{  
    if (check(status[i/32], i%32) == 0)  
        prime.pushback[i];  
    }  
}
```

## How works

Suppose  $i = 3$

now  $i/32 = 3/32 = 0$   $\text{status}[0]$  index 0

$i \mod 32 = 3 \mod 32 = 3$  pos: 3  $\rightarrow$  3rd bit

\* check ( $\text{status}[\frac{3/32}{0}]$ ,  $\frac{3 \mod 32}{3}$ )

$\rightarrow$  [index] value = 0  
... in binary index value = 0 (binary)

$$\rightarrow [1 \ll \text{position}] = 1 * 2^{\text{pos}}$$

$$= 1 * 2^3$$

$$= 8$$

= 1000 (in binary)

now [index & C/K position]

$$\rightarrow 0 \& 1000$$

$$\rightarrow 0$$

and

1	0	0	0	0
*	1	0	0	0
-----	0	0	0	0

$\therefore \text{ans} == 0$ ; 3 is prime number

now mark all its multiples

$$J = i \times i \rightarrow J = 3 \times 3 = 9$$

$$J = 9$$

~~status[9/32] = something~~  
~~index~~

$$9 \cdot 32 = 3 \cdot 9 \rightarrow \text{position}$$

~~status[9/32] =~~

set (status[9/32], 9)

$$\rightarrow \text{Index} = 0 \\ = 0 \text{ (binary)}$$

$$\rightarrow 1 \ll \text{position} = 2 * 2^9$$

$$= 512 \\ = 100\ 000\ 000 \text{ (binary)}$$

$$\rightarrow \text{index } 1 \text{ (} 1 \ll \text{position} \text{)} \rightarrow 0 \mid 1000\ 00000 \\ \rightarrow 1000\ 00\ 000$$

$$\therefore \text{index} = 512$$

$$\therefore \text{status}[9/32] = 512 \text{ (settled)}$$

$i = 7$

$7 \& 32 = 2 \rightarrow \text{status}[2] \rightarrow \text{index } 2$

$7 \& 1 \cdot 32 = 7 \rightarrow \text{pos} \rightarrow 7$

# check ( 2 , 7 )

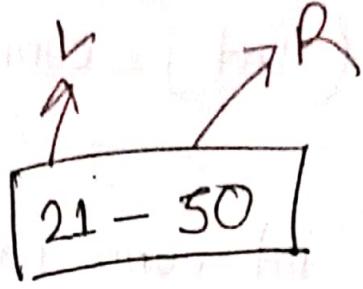
$\therefore \text{index } 2 = 10 \text{ (binary)}$

$\therefore 1 \ll \text{position} = 2^7 = 128$   
 $= 1000000.0 \text{ (binary)}$

$\therefore \text{Index } \delta \text{ & } 1 \ll \text{position} = 10 \text{ & } 1000000.0$   
 $= 0$

$\therefore \boxed{71 \text{ is prime}}$

segmented  
sieve



vector<int> prime;  
sieve();

void segsieve( int L , int R )

C  
vector<bool> isprime(  $\lceil \frac{R-L+1}{\text{prime}} \rceil$ , true )<sup>Initial all true</sup>

if ( $L == 1$ ) isprime[0] = false  $\rightarrow$  extreme case

for (int i=0 ; prime[i]\*prime[i] <= R ; i++ )

② int current prime = prime[i]

int currbase = (L / current prime)  $\downarrow$  current prime  
20  $\downarrow$  21  $\downarrow$  2

$\therefore$  if (currbase < L) currbase += current prime  
20  $\downarrow$  21

(22)

for (int j = currbase ; j <= n ; j += currprime)

$$\{ \text{int currIndex} = j - L \}^* \text{index}$$

isprime[currIndex] = false

}

(After all the J assigned by

if (currbase == currprime)

isprime[currbase - L] = true

for (int i = 0 ; i < (n-1+1) ; i++)

{ if(isprime[i])  
cout << i << endl.

}

1

2  
Prime  
plant

3<sup>2nd</sup>

4

5<sup>3rd</sup>

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21  
X

22  
X

23

24  
X

25  
X

26  
X

27  
X

28  
X

29

30  
X



prime  
in Range

21 - 30

## Bellman Ford

→ complexity  $O(VR)$

### basic pseudocode

- ① Initial all distance [vertex] = INT-max  $\infty$
- ② Except source node Distance [source] = 0
- ③ Relax all edges (node-1) times

for( node-1 )

  for( edges )

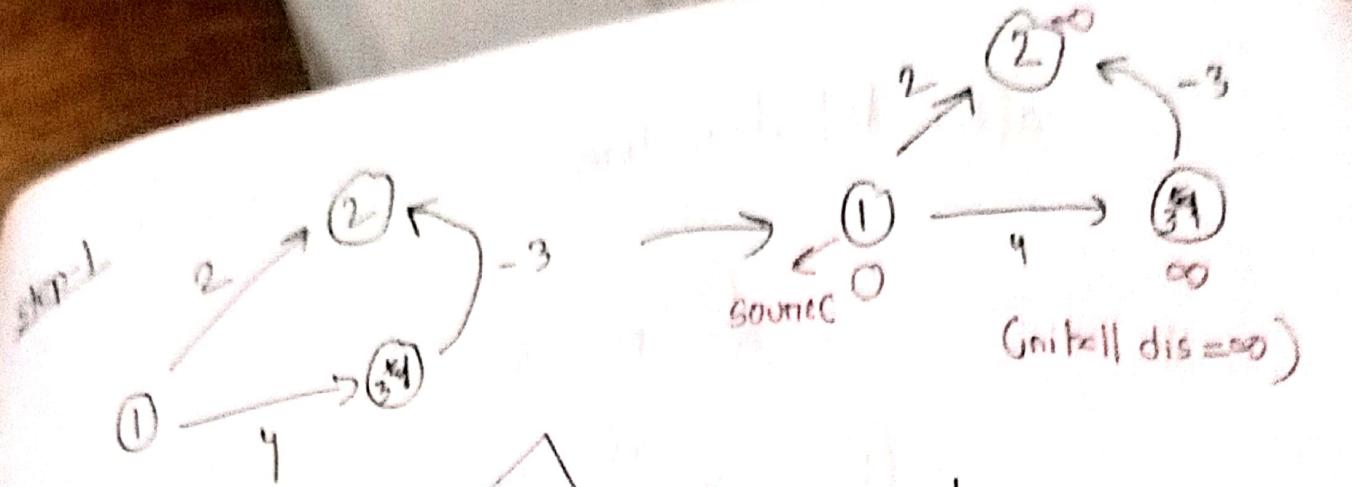
    if  $dis[v] > dis[u] + \text{weight}$

$\therefore dis[v] = dis[u] + \text{weight}$

Relaxing

$v \xrightarrow{\text{wt}} v$

- ④ Negative weight cycle Detection



list of edge

3 2 ✓

1 3 ✓

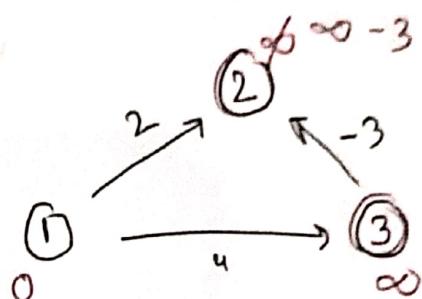
1 2

any order

adj List

nodes = 3  
 $(3-1)/2$  times  
 iteration  
 $(3, 2)$   
 $(v, u)$

↓ first iteration

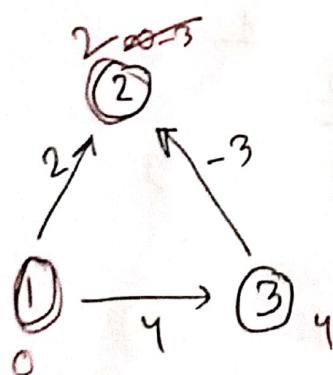


$$dis[v] > dis[u] + wt$$

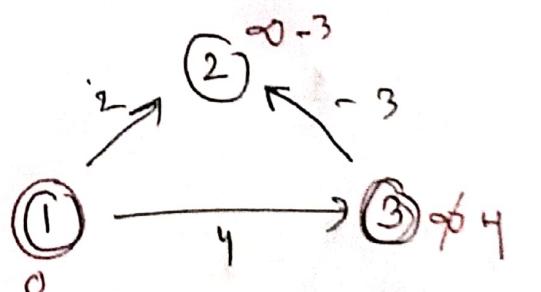
$$\therefore dis[v] = dis[u] + wt$$

$$\begin{aligned} dis[2] &= dis[3] + wt \\ &= \infty - 3 \end{aligned}$$

(1, 2)



(1, 3)

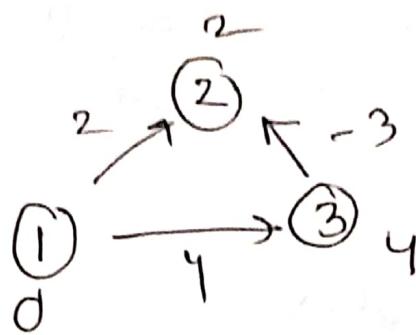


$$dis[2] > dis[1] + 2$$

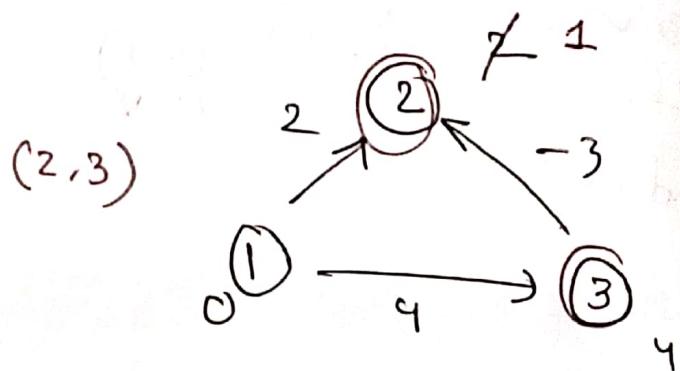
$$\therefore dis[2] = 2$$

$$\begin{aligned} \text{as } dis[3] &> dis[1] + 4 \\ \therefore dis[3] &= 4 \end{aligned}$$

After 1st iteration



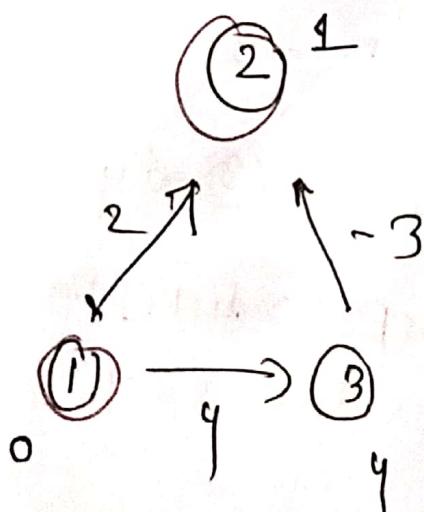
↓ (2nd iteration)



$$\text{dis}[2] > \text{dis}[3] - 3$$

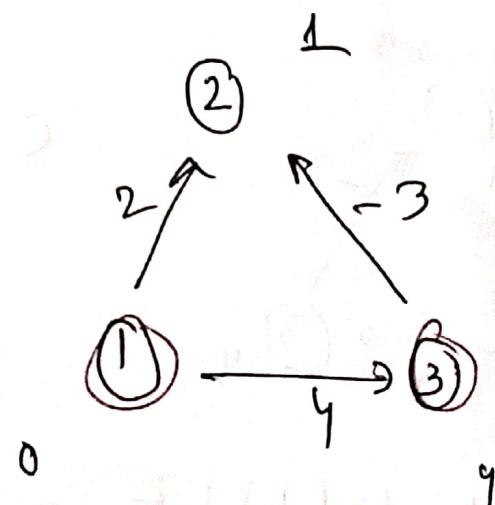
$$\text{dis}[2] = 1$$

(1, 2)



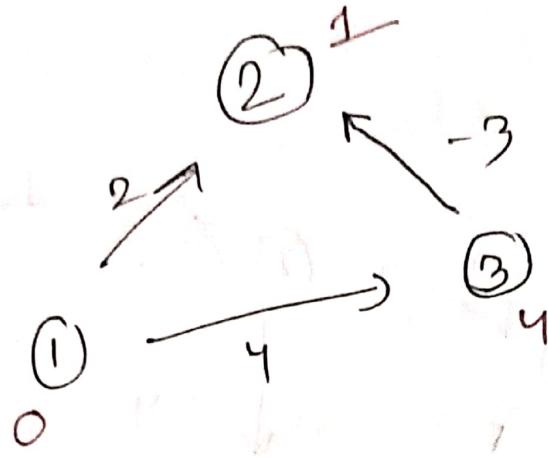
Same

(1, 3)

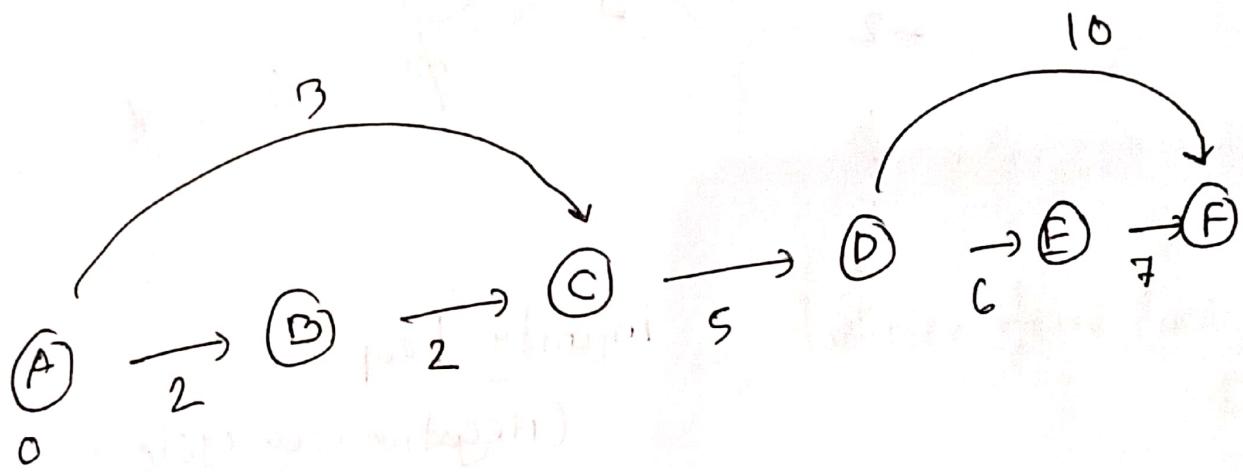


(same)

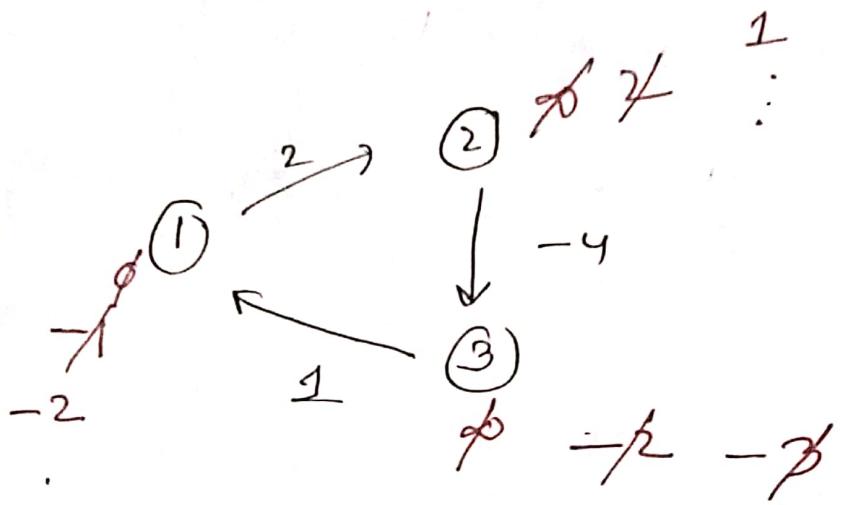
after all iteration graph is now



1	2	3
0	1	4



Negative  
wt cycle



infinity Loop  
(Negative weg cycle)

```
int main()
{
    node, edge ;
    vector<vector<int>> adj ;
    for( int i=0 ; i < edge ; i++)
    {
        node1, node2, weight
        adj.pushback({node1, node2, weight})
    }
}
```

vector<int> dist = bellman\_ford(node, 1, adj)

```
vector<int> bellmanford (int node, int source,  
vector<vector<int>> adj)
```

# initial all  $\infty$  distance

```
vector<int> dis (node+1, INT_MAX)
```

```
[vector<int> parent (node, 0)]  $\infty$ 
```

```
dis[source] = 0;
```

# Relaxing nodes/edges

```
for ( i = 0 to node - 1 )
```

```
{
```

```
for (auto edge : adj)
```

```
{
```

```
    node1 = edge[0]      wt = edge[2]
```

```
    node2 = edge[1] +
```

```
if (distance [node1] + wt < dis [node2])
```

```
    dis [node 2] = dis [node 1] + wt
```

```
[parent [node2] = node1]
```

```
y
```

~~W~~ often doing 1 more relaxation if any distance get  
change that means graph  $\rightarrow$  negative wt cycle

for (auto edge : adj)

{  
node1, node2, a edge wt

: if (dis[nodej] > dis[node1] + wt)

{  
"Negative wt cycle"  
g exit(0);

}

# parent printing / path

int presentnode = node

vector<int> path

while ( parent[presentnode] != 0 )

path.pushback(presentnode)

presentnode = parent[presentnode]

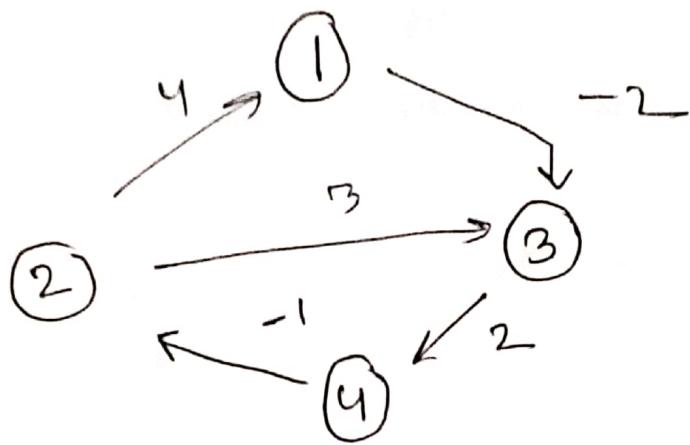
reverse (path.begin(), path.end())

for (auto node : path)

cout < node

# Floyd Warshall

$\rightarrow O(N^3)$  time

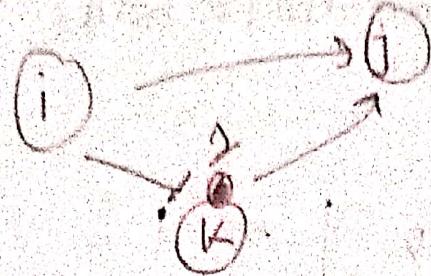


adj List  
↳ dis-mat

	1	2	3	4
1	0	$\infty$	-2	$\infty$
2	4	0	3	$\infty$
3	$\infty$	$\infty$	0	2
4	$\infty$	-1	$\infty$	0

$0 \xrightarrow{-2} 3$   
 $1 \xrightarrow{4} 1$   
 $2 \xrightarrow{3} 3$   
 $3 \xrightarrow{2} 4$   
 $4 \xrightarrow{-1} 2$

gas intermediate vertex



update path

	1	2	3	4
1	0	$\infty$	-2	$\infty$
2	4	0	$\frac{3}{2}$	$\infty$
3	$\infty$	$\infty$	0	$\infty$
4	$\infty$	-1	$\infty$	0

1

as 1  
1st now  
and col  
notches)

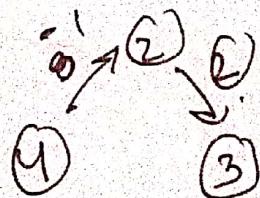
$$d_{ij} = \min(d_{ij}(u) + d_{kj}(j))$$

2 as intermediate

3nd

	1	2	3	4
1	1	0	$\infty$	-2
2	3	4	0	$\infty$
3	$\infty$	$\infty$	0	2
4	9	-1	$\infty$	0
	$\varnothing_3$		$\varnothing_{\varnothing}$	
			<del>2</del>	
			1	

check dis  
from table  
not graph



1

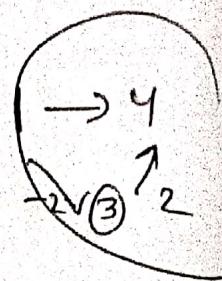
4

$$4 \xrightarrow{1} 3$$

3rd as intermediate node

(9)

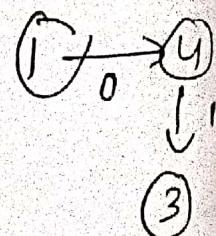
	1	2	3	4
1	0	<del>0</del>	-2	0
2	4	0	2	$\frac{9}{2}$ $2+2=4$
3	$\infty$	$\infty$	0	2
4	<del>3</del>	-1	1	0



4 as intermediate node

(9)

	1	2	3	4
1	0	<del>0</del> $0+(-1)=-1$	-2	0
2	4	0	2	4
3	$\frac{9}{2}$ $2+3=5$	<del>0</del> <del>2+3=5</del>	0	2
4	3	-1	1	0



	1	2	3	4
1	0	-1	-2	0
2	u	0	2	4
3	5	1	0	2
4	3	-1	1	0

1st check  
gnph mentable

if possible

(1) → (2) → (3) X not work  
 $\nwarrow$  (3) ← (2) ✓ overlap doesn't work