

Agenda

1) Bit Manipulation \rightarrow Binary \rightarrow 0 & 1

↳ Divide Two Integers ✓

2) Reversal in Linked List ✓

3) cycle Detection in Linked List ✓

↳ Floyd's cycle detection

Bit Manipulation

(0-9)

$$\begin{aligned}
 (204)_{10} &= 2 \times 10^2 + 0 \times 10^1 + 4 \times 10^0 \\
 &= 200 + 0 + 4 \\
 &= 204
 \end{aligned}$$

(0-1)

$$\begin{aligned}
 (101)_2 &= 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 \\
 &= 1 + 0 + 4 \\
 &= 5
 \end{aligned}$$

5 \rightarrow Binary Number

$$\begin{array}{r|rrr}
 2 & 5 & & \\
 \hline
 2 & 2 & 1 & \Rightarrow 101 \\
 & 1 & 0 &
 \end{array}$$

significance

Left shift ($<<$)

$$a = 101$$

multiplication

by 2

$$\begin{array}{l} a * 2 \\ a \ll 1 \\ \hline \text{bits} \end{array}$$

$$a \ll 1$$

$$\begin{array}{r} 101 \\ \hline 1 \ 0 \ 1 \ 0 \\ \uparrow \uparrow \uparrow \uparrow \\ 3 \ 2 \ 1 \ 0 \\ 0 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + \\ \hline 1 \times 2^3 \\ \Rightarrow 10 \end{array}$$

- a) 0101
 b) 1010
 c) 011
 d) None of
 these

$$\begin{array}{r} 1 \ 0 \ 1 \ 0 \\ \uparrow \uparrow \uparrow \uparrow \\ 4 \ 3 \ 2 \ 1 \ 0 \\ 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + \\ \hline 0 \times 2^3 + 1 \times 2^4 \end{array}$$

$$4 + 16 = 20$$

significance

Right shift (\gg)

Division by 2

$$a \gg 1$$

$$a = 101$$

$$010$$

$$\underline{2}$$

$$5 / 2 = 2$$

$$\begin{array}{r} 101 \\ \hline 010 \\ \underline{001} \\ \hline 1 \end{array}$$

$$1 \times 2^0 = 1$$

Note → Left shift → multiplication by 2

Right shift → Divide by 2

Check whether a is even or odd ??

AND operation with 1

Marking

$a = 24$

\hookrightarrow if $a \% 2 == 0$ even
else: Odd

$1 \rightarrow 001$

$$\begin{array}{r} 100 \\ 001 \text{ (AND) } \wedge \\ \hline 000 \end{array} \rightarrow 0$$

even

\hookrightarrow Last digit is 0

$$\text{Even} \rightarrow 4 = 100$$

$$\text{Odd} \rightarrow 5 = 101$$

$$\begin{array}{r} 101 \\ 001 \text{ (AND) } \wedge \\ \hline 001 \end{array}$$

num

\hookrightarrow Last digit is 1

↓
odd
num

AND

$$\begin{array}{r} x \\ 1 \\ 1 \\ 0 \\ 0 \end{array} \quad \begin{array}{r} y \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{array} \quad \begin{array}{l} \text{O/P} \\ \hline 1 \\ 0 \\ 0 \end{array}$$

%, /, *

Divide two integers

$47/4 = 11$

$$\left\{ \begin{array}{l} \text{Dividend} = 47 \\ \text{Divisor} = 4 \end{array} \right. \quad \underline{\text{Output} = 11}$$

while (dividend - divisor) > divisor:

 count += 1

return count

Repeatedly doing subtraction

47 - 4	43 - 4	39 - 4	35 - 4	31 - 4	27 - 4	23 - 4
43	39	35	31	27	23	19
1	2	3	4	5	6	7

19 - 4	15 - 4	11 - 4	7 - 4	
15	11	7	3	
8	9	10	11	
			=	

output = 11

dividend / divisor

Time complexity = $\Theta(n)$

Space complexity = $\Theta(1)$

$$\left\{ \begin{array}{l} \text{dividend} = 2^{31} \\ \text{divisor} = 2 \end{array} \right. \rightarrow \text{Time Limit Exceeded}$$

Drawback → Range b/w (TLE)
Dividend &

Divisor → too high

Left Shift

$$\text{Dividend} = 47 \quad 4 \quad \underline{\hspace{2cm}} \quad (2^0)$$

$$8 \quad \underline{\hspace{2cm}} \quad (2^1)$$

$$\text{Divisor} = 4 \quad 16 \quad \underline{\hspace{2cm}} \quad (2^2)$$

$$\text{Result} = 0 \quad 32 \quad \underline{\hspace{2cm}} \quad (2^3)$$

$$\text{shift} = 0 \quad | 6 \quad \sum = 16 \quad \underline{64} \quad \underline{\hspace{2cm}} \quad (2^4)$$

while (dividend \geq (divisor \ll shift)):

$$\text{Inside loop} \quad \text{shift} += 1 \quad \text{shift} = 4$$

$$\text{result} += (1 \ll (\text{shift}-1))$$

$$= 1 \ll (3) = 2^3 = 8$$

$$\text{dividend} -= \text{divisor} \ll (\text{shift}-1)$$

$$\text{Dividend} = \underline{\hspace{2cm}} \quad 47 - 32 = 15$$

$$\text{Divisor} = 4 \quad \underline{\hspace{2cm}} \quad (2^0)$$

$$\underline{8} \quad \underline{\hspace{2cm}} \quad (2^1)$$

$$16 \quad \underline{\hspace{2cm}} \quad (2^2)$$

$$\text{shift} = 2$$

$$\text{Result} = 8 + (1 \ll ^{2^1}(\text{shift} - 1))$$

$$\text{Result} = 10$$

$$\text{dividend} = 15 - 8 = 7$$

$$\text{divisor} = \frac{4}{8} \quad (2^0) \quad (2^1) \quad \text{Shift} = 1$$

$$\text{Result} = 10 + 1 = 11$$

$$\text{dividend} = 7 - 8 = -\text{ve}$$

Example 2

$$\text{Dividend} = 48$$

$$\text{Output} = 12$$

$$\text{Divisor} = 4$$

$$\text{Dividend} \Sigma = (\text{Divisor} \ll \text{shift})$$

$$\left\{ \begin{array}{l} 4 \quad (2^0) \\ 8 \quad (2^1) \end{array} \right.$$

$$2^3$$

$$\downarrow$$

$$1 \ll (\text{shift} - 1)$$

$$\left\{ \begin{array}{l} 16 \quad (2^2) \\ \rightarrow (32) \quad (2^3) \end{array} \right.$$

$$\text{Result} = 0 + 2^3$$

$$= 8$$

$$\begin{array}{r} 64 - \\ \hline \end{array} \quad (2^4)$$

$$\text{divid} = 48 - 32$$

$$= 16$$

dividend = 16

divisor = 4

$$\begin{array}{r} 4 \quad (2^0) \\ 8 \quad (2^1) \\ \boxed{16} \quad (2^2) \\ \hline 32 \quad (2^3) \end{array}$$

result = $8 + 2^2$
 = 12

$$\text{dividend} = 16 - 16 = \underline{0}$$

○ ≠ 4 → stop

$$\text{dividend} = \text{abs}(\text{dividend})$$

$$\text{divisor} = \text{abs}(\text{divisor})$$

$(\text{dividend} < 0 \text{ and divisor} \geq 0) \text{ OR}$

$(\text{divisor} < 0 \text{ and dividend} \geq 0)$

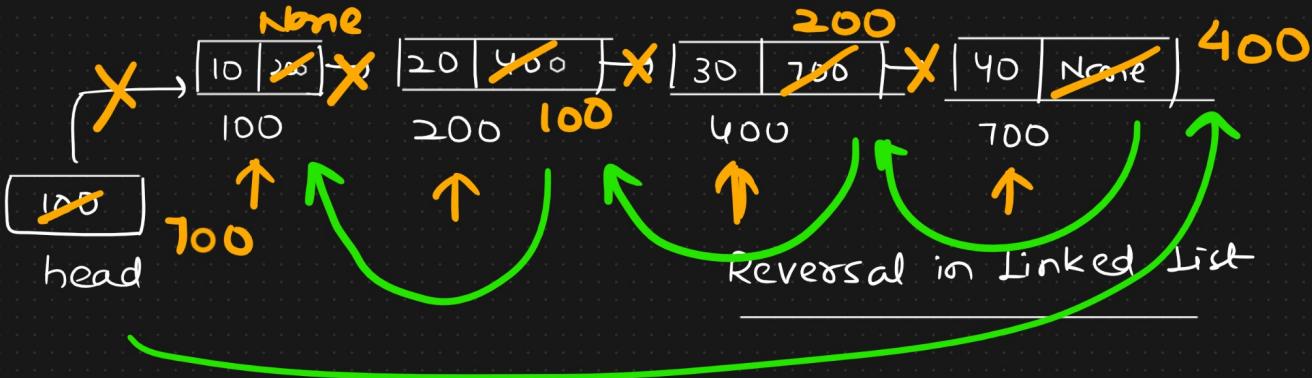
$$\text{result} = - \text{result}$$

$$\underline{\text{next}} = \text{curr.next} - 1$$

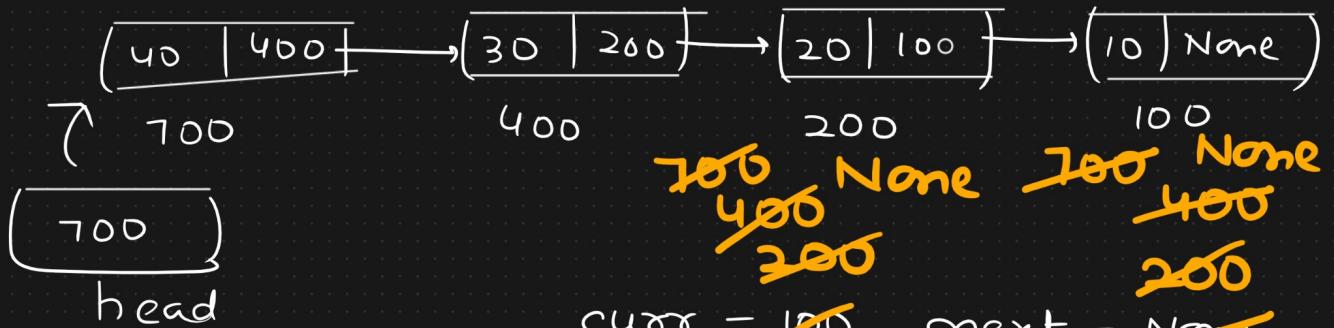
$$\text{curr.next} = \text{prev} - 2$$

$$\text{prev} = \text{curr} - 3$$

curr = next
 4



Expected output



curr = self.head

prev = None

while curr:

 next = curr.next ✓

 curr.next = prev ✓

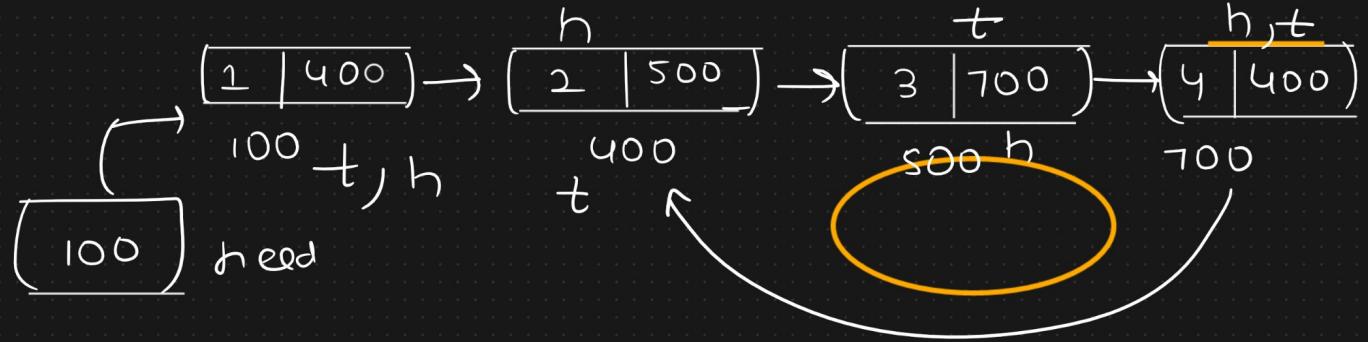
 prev = curr ✓

 curr = next ✓

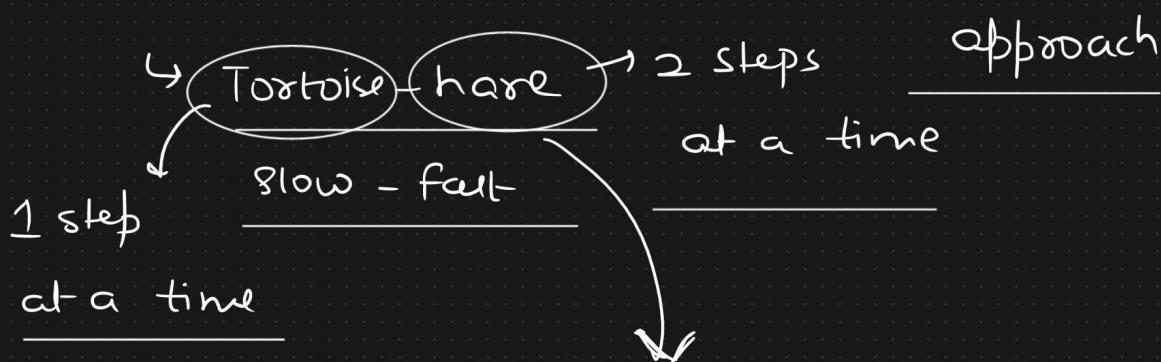
 100
 200
 300
 400
 700

self.head = prev

Meeting Point



Floyd's cycle detection algo → Two Pointers

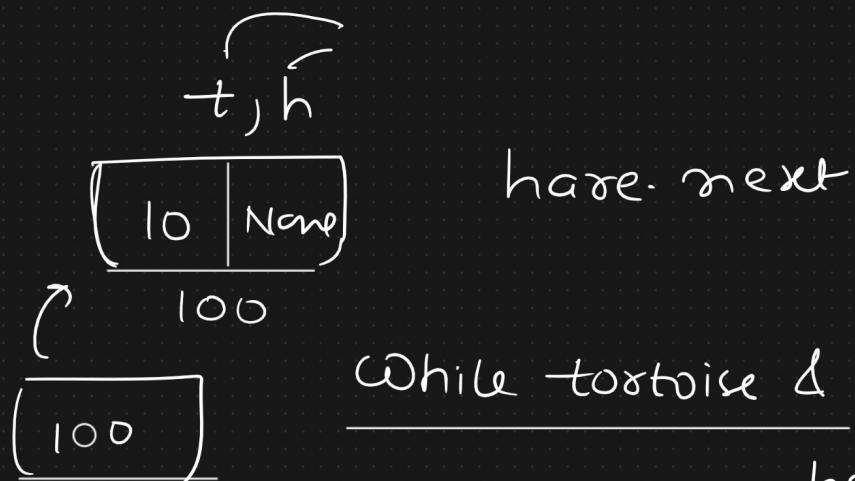


tortoise & hare meet

↳ Loop / cycle

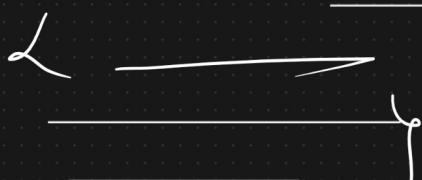
Otherwise

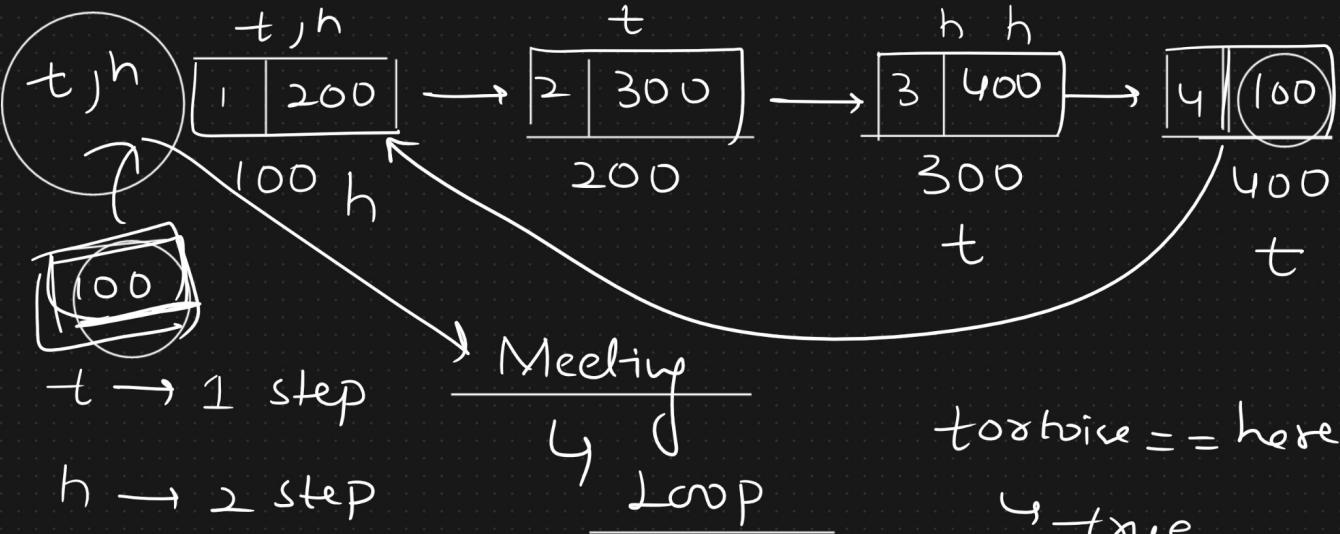
↳ No cycle



While tortoise & hare &

hare.next :





two-pointers
approach

$\text{to twice} == \text{here}$

true

false