1 → 2 → 2 → 1

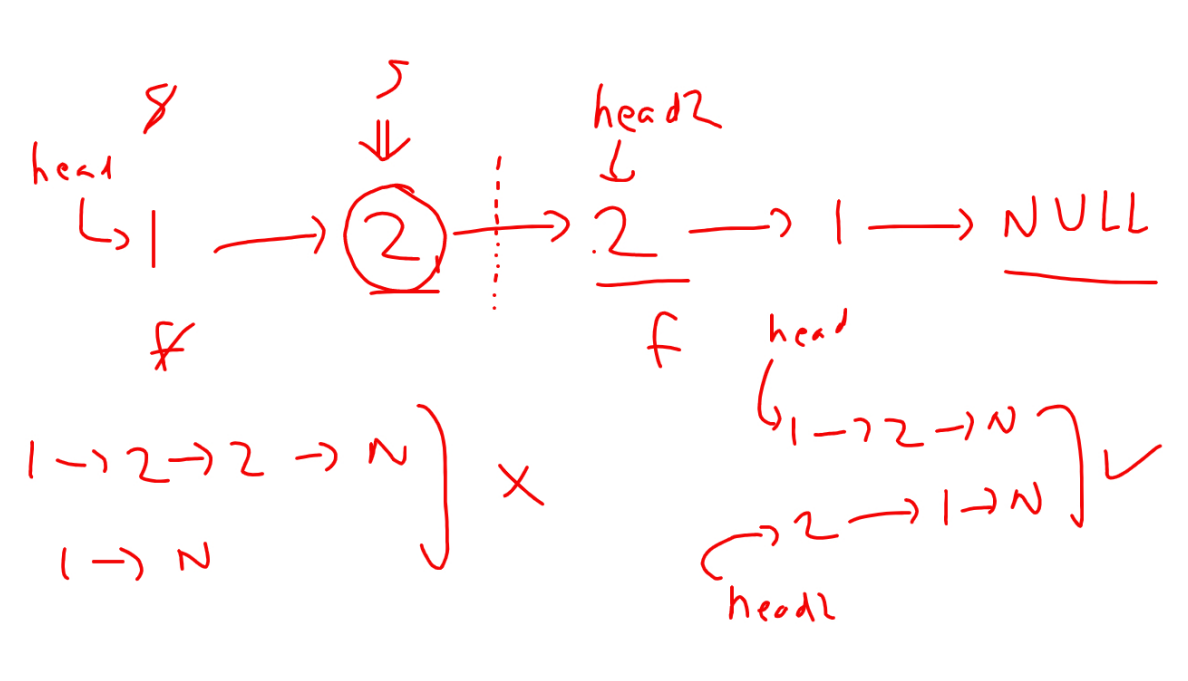
| 1 | 2 | 2 | 1 |

O(n) aux space in rec. space.

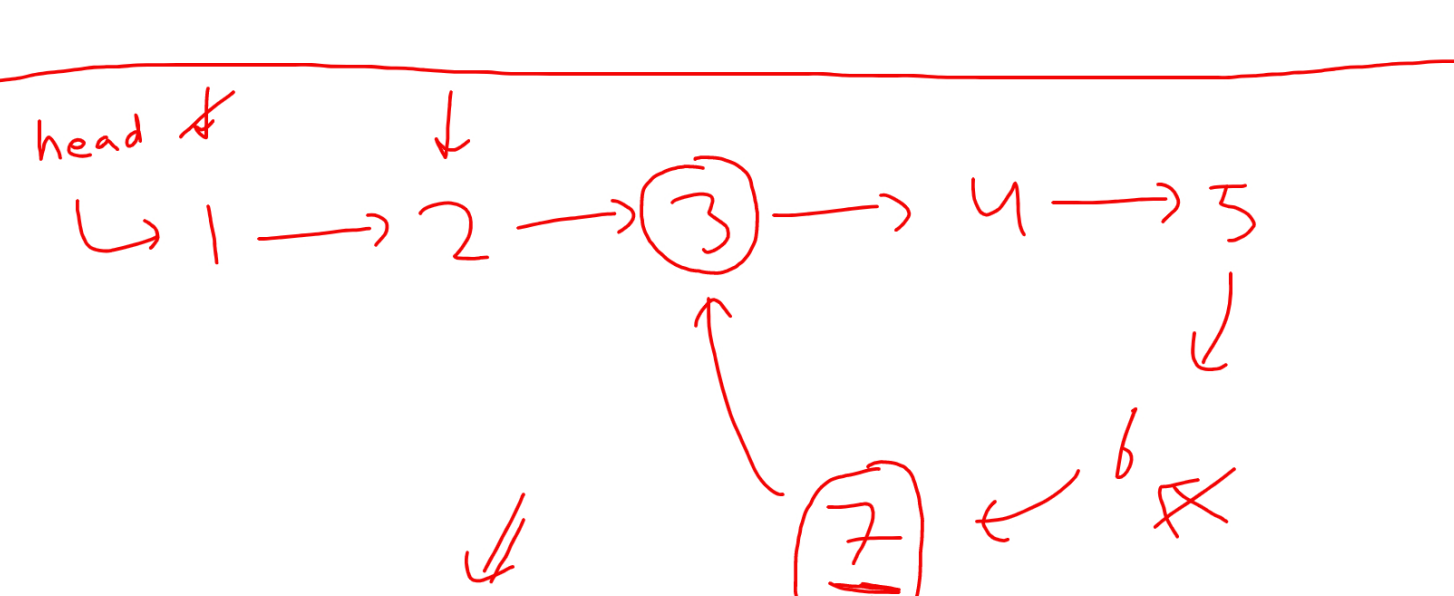
[1 → 2 → NULL]
2 → 1 → NULL ⇒ (Rev) [1 → 2 → NULL]

1 → 2 → 3 → 2 → 1 → NULL

[1 → 2 → 3 → NULL]
2 → 1 → NULL ⇒ (rev) [1 → 2 → NULL]



1 → 2 → N
1 → 2 → N



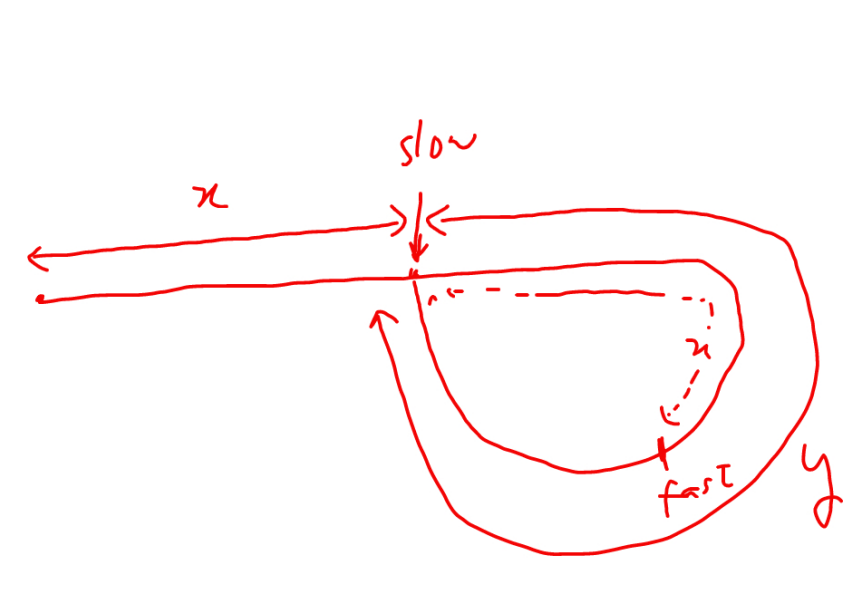
1 → 2 → 3 → 4 → 5 → 6 → 7 → NULL [3x]

[address → Boolean]
1x → True
2x → True
[3x → T]
4x → T
5x → T
6x → T

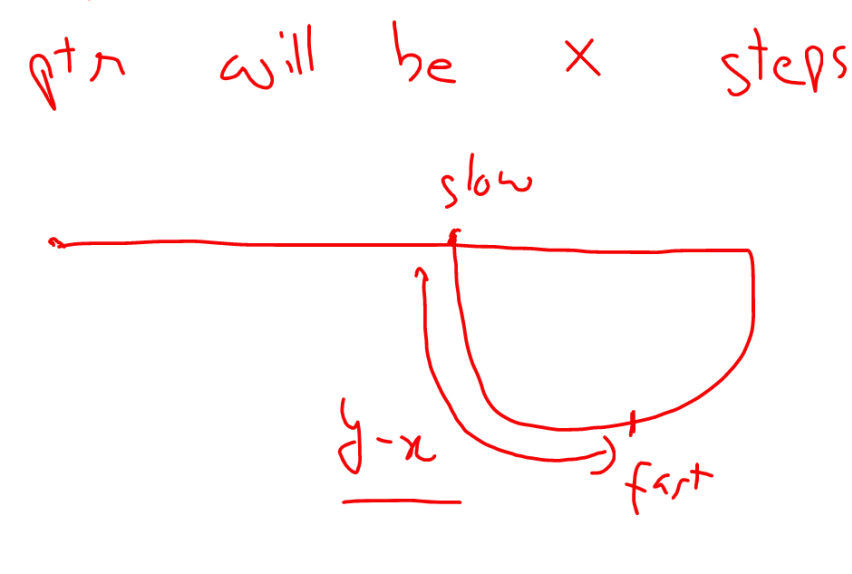
slow : One step at a time
fast : Two steps at a time

Claim: If the L.L. contains a cycle, the slow & fast pointers will eventually meet somewhere.

Proof:

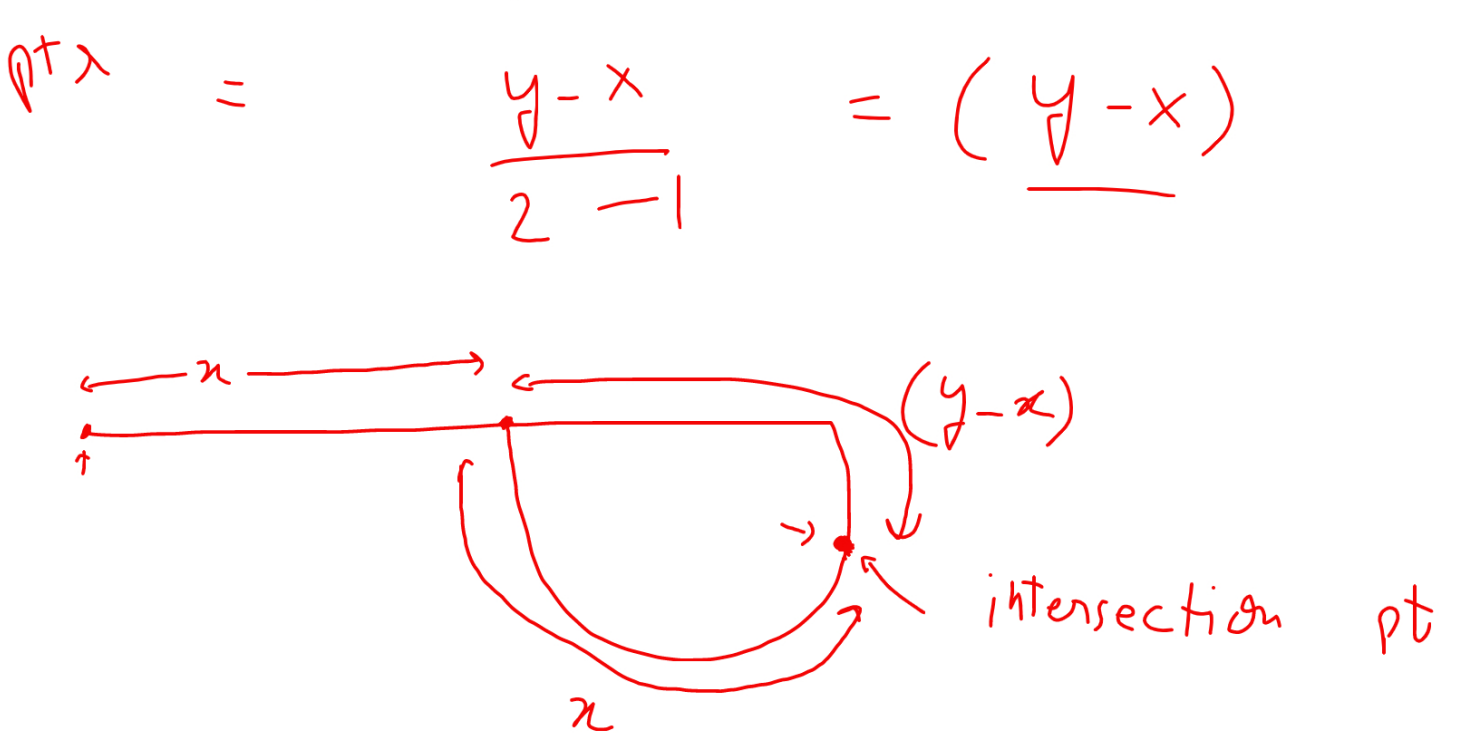


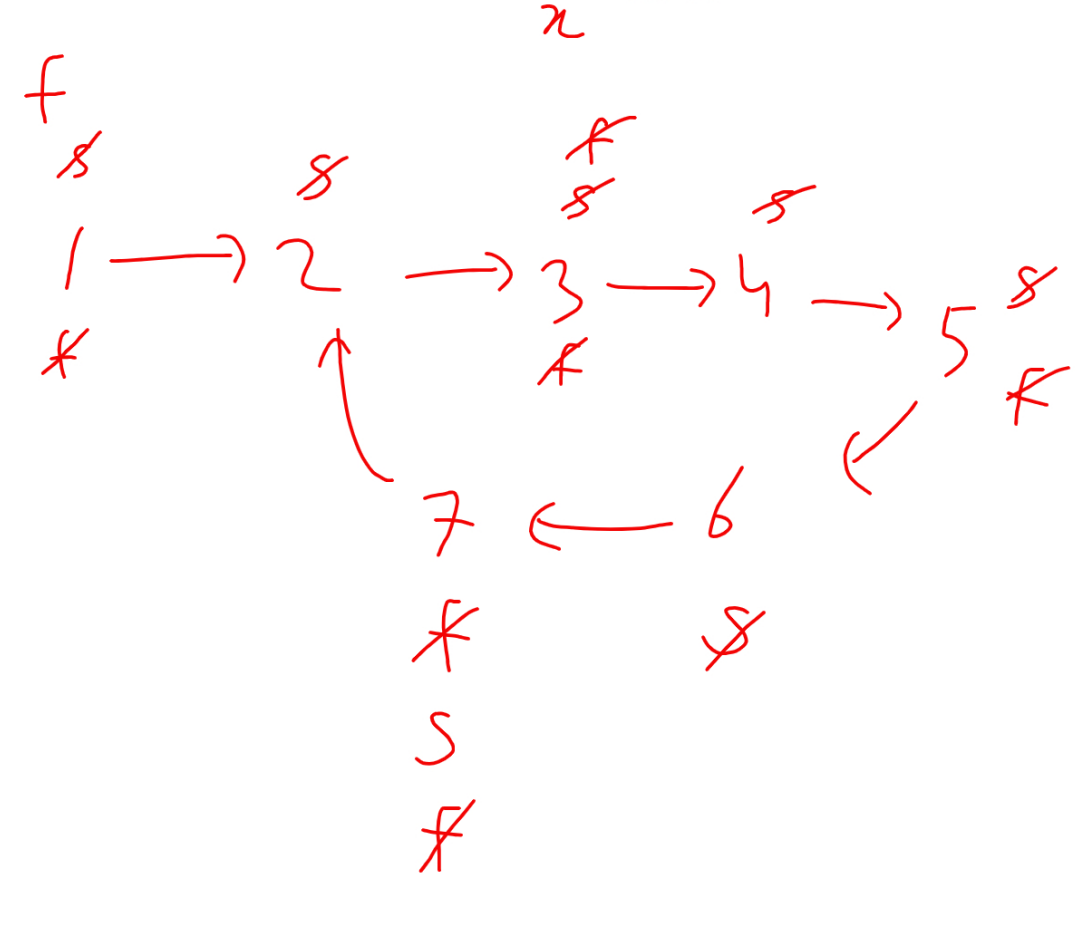
⇒ when 'slow' ptr reaches the intersection point, then the 'fast' ptr will be x steps inside the loop

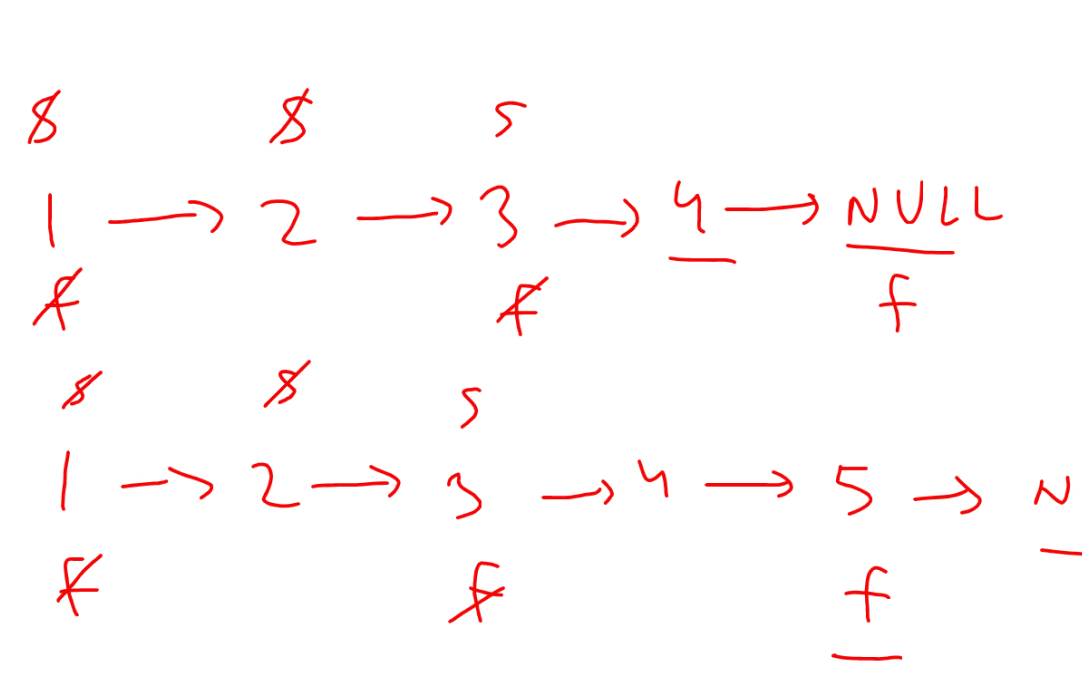


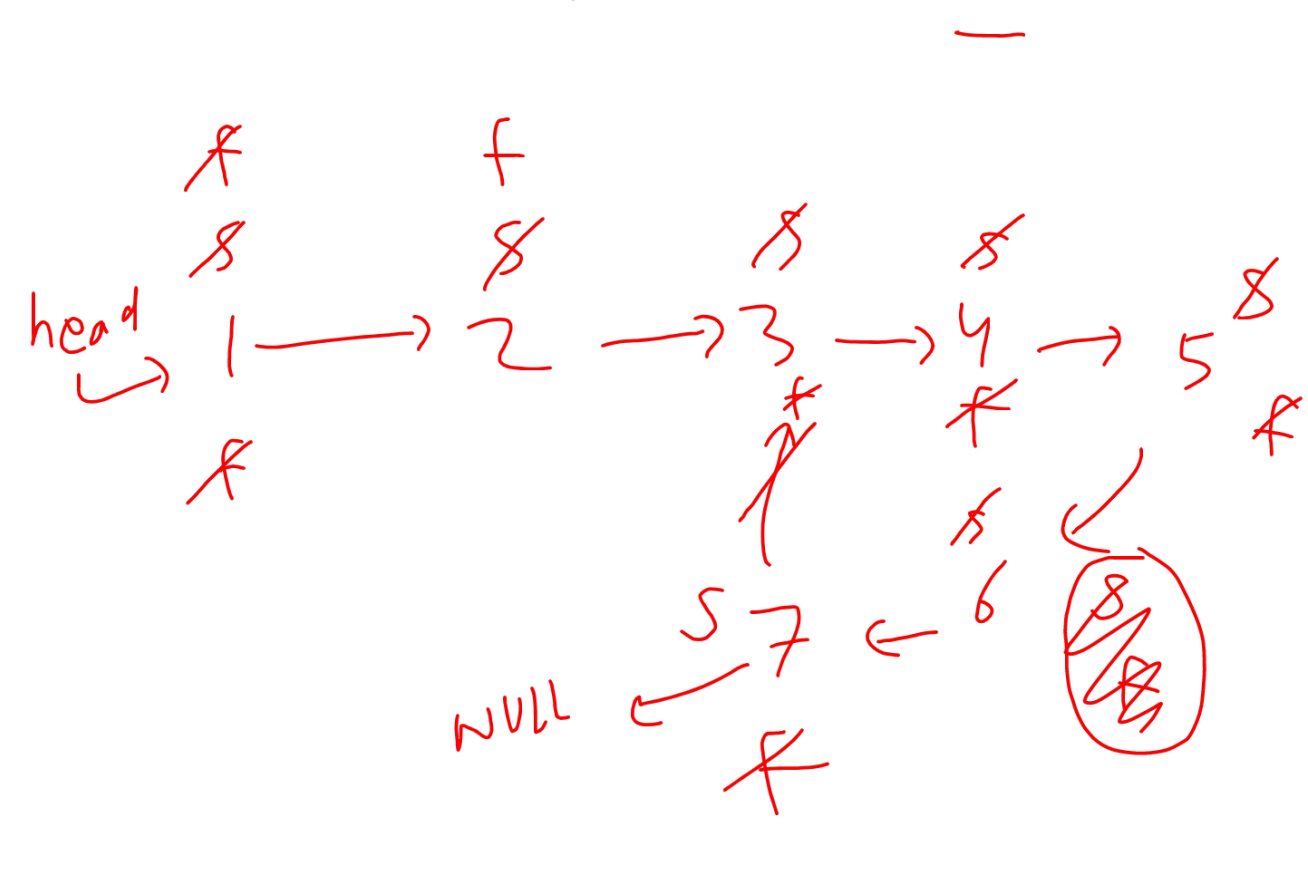
⇒ ['fast' ptr is (y−x) steps behind the 'slow' ptr.
Speed of 'fast' ptr = 2
Speed of 'slow' ptr = 1]

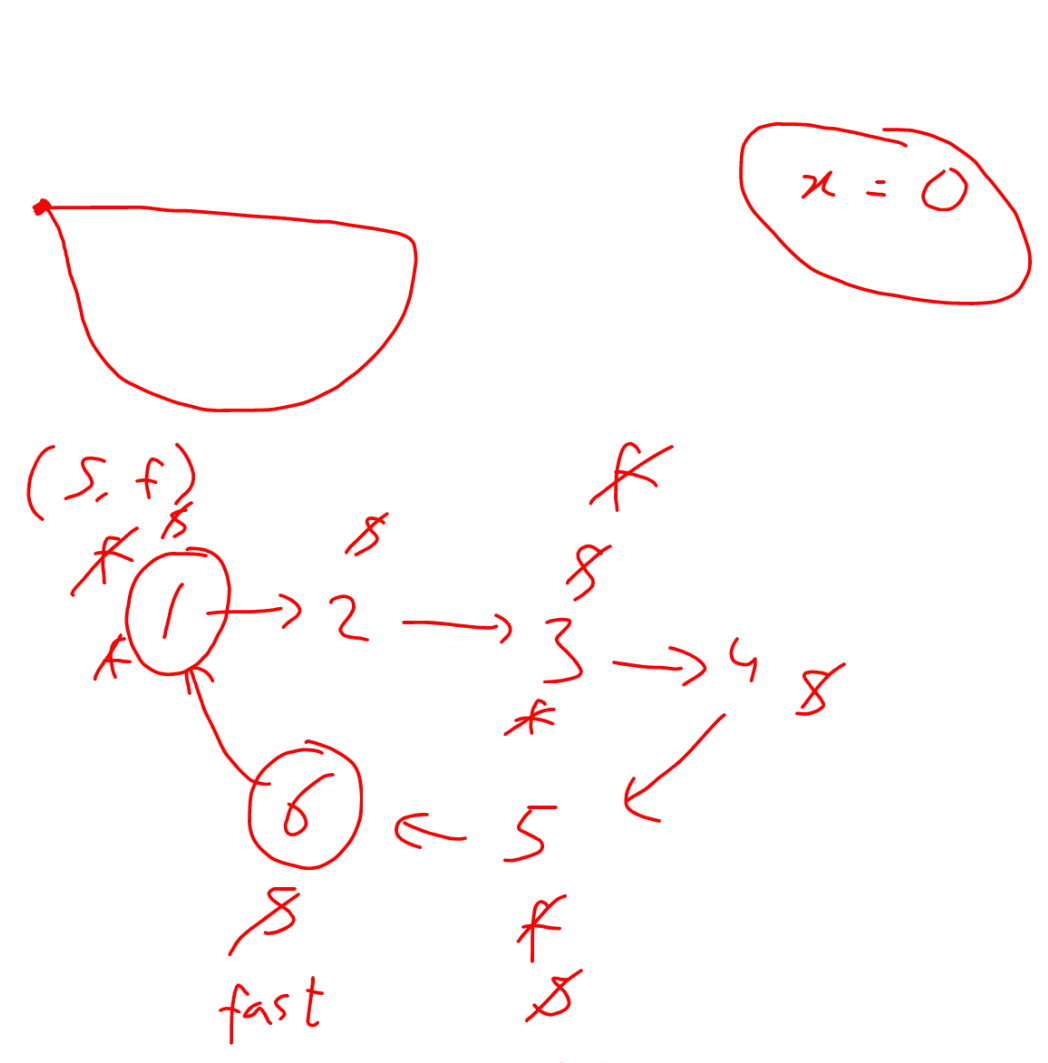
time need for 'fast' ptr to catch the 'slow' ptr $= \frac{y-x}{2-1} = (y-x)$

⇒










head
→ 1 → NULL
X

1 → 2 → 3 → 4 → 5 → 6 → NULL ✓

Binary Representation:

0/1

[decimal repr. = 14]

[Binary repr.]

2	14
2	7
2	3
2	1
	0

remainders
0
1
1
1

Binary repr. of 14 = 1 1 1 0 ($\dots 2^3\ 2^2\ 2^1\ 2^0$)

$(0\times2^0) + (1\times2^1) + (1\times2^2) + (1\times2^3)$
$= 0 + 2 + 4 + 8 = 14$

Bitwise Operators:

OR (|)
AND (&)
XOR (^)
NEGATE (~)

OR:

I₁	I₂	O/P
0	0	0
0	1	1
1	0	1
1	1	1

AND:

I₁	I₂	O/P
0	0	0
0	1	0
1	0	0
1	1	1

XOR

I₁	I₂	O/P
0	0	0
0	1	1
1	0	1
1	1	0

NEGATE

I/P	O/P
0	1
1	0

Left shift (<<) by 1: 0 0 0 1 0 1 0
0 0 1 0 1 0 0 (Multiplies the I/P by 2)

Right shift (>>) by 1: 0 0 0 1 0 1 0
0 0 0 0 1 0 1 = (5) (Divides the I/P by 2)
0 0 0 0 0 1 0 = (2)

If the rightmost bit of a number is set, then the number will be:
A) Even
B) Odd
C) Can't say

$\underbrace{\_\ \_\ \_\ \_\ 2^3\ 2^2\ 2^1}_{\text{Even weights}}\ \underbrace{2^0}_{\text{odd weight}}$

Even + Even = Even
Even + Odd = Odd

XOR
[a ^ a = 0
0 ^ a = a] IMP

0 → 0 0 0 0 0 0 0
a → 0 1 1 0 1 0
0 1 1 0 1 0 = (a)

# Given a list of numbers where each no. occurs twice except one number. Find that single number.

[1, 2, 3, 2, 1, 5, 5] O/P = 3

$1\text{^}2\text{^}3\text{^}2\text{^}1\text{^}5\text{^}5 = \underbrace{1\text{^}1}\text{^}\underbrace{2\text{^}2}\text{^}\underbrace{5\text{^}5}\text{^}3$
$= 0\text{^}0\text{^}0\text{^}3$
$= 0\text{^}3 = \boxed{3}$

---

10 : 1 0 1 0 ($8\ 4\ 2\ 1$) O/P = 2
15 : 1 1 1 1 O/P = 4

(n & 1) == 1 ⇒ Rightmost bit is set.
==0 ⇒ " " unset

discard the rightmost bit.

$$\begin{array}{r} 1010 \\ \&\ 0001 \\ \hline 0 \end{array}$$ count = 0

$$\begin{array}{r} 0101 \\ \&\ 0001 \\ \hline 1 \end{array}$$ count = 1

$$\begin{array}{r} 0010 \\ \&\ 0001 \\ \hline 0 \end{array}$$ count = 1

$$\begin{array}{r} 0001 \\ \&\ 0001 \\ \hline 1 \end{array}$$ ⇒ count = 2

$\boxed{0000}$ STOP

n>>1

Check whether the kth bit is set or not.

n = 10
k = 3

1 0 1 0 ($3\ 2\ 1\ 0$)

Left shift
0 0 0 1
0 0 1 0
0 1 0 0
1 0 0 0

$$\begin{array}{r} 1010 \\ 1000 \\ \hline 1000 \end{array}$$

Count set bits in O(number of set bits) time.

n = 10
n−1 = 9

$$\begin{array}{r} 1010 \\ \&\ 1001 \\ \hline 1000 \end{array}$$ ($8\ 4\ 2\ 1$) count = 1

1+8

n = 1 0 0 0
n−1 = & 0 1 1 1
$\boxed{0000}$ = 0 STOP

count = 2

n = 1 0 1 0 0 1 0 0 0
n−1 = 1 0 1 0 0 0 1 1 1
1 0 1 0 0 0 0 0 0

Keep doing [n = n & (n − 1)] till n becomes eq. to 0.

[1, 2, 3]

[ ]
[1] [2] [3]
[1, 2] [1, 3] [2, 3]
[1, 2, 3]

= 8 subsets

[2, 1]

[2, 1]   [1, 2]

⇒ [1, 2, 3]
$2 \times 2 \times 2 = 2^3$

[1 2 3 4]
$2 \times 2 \times 2 \times 2 = 2^4$

n 2^n

8

i	3	2	1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

0 1 2
[1 2 3]

[ ]
[1]
[2]
[1, 2]
[3]
[1, 3]
[2, 3]
[1, 2, 3]

[2 × 2 × 2]

n 2^n

i : [0 → 2^n − 1]

i = 5 ⇒ num = 5

1 0 1   pos = 0
0 1 0   pos = 1
0 0 1   pos = 2
0 0 0

[1, 3]