Generated by Bing AI image creator
https://www.bing.com/images/create
Query: “linked list code algorithm visualization”

Topics 8
1_mid element in LL ✓
✓ 2_ merge two sorted LL
3_ merge sort LL
4_cycle?
5_begining of cycle

P1 Find middle element in a Linked List. ret node pointer

ex1 head → 2 → 5 → 10 → 3 → 18 → null
0 1 2 3 4
↑ ans

ex2 head → 2 → 5 → 10 → 3 → 18 → 18 → null
0 1 2 3 4 5
↑ ans

idea 1

2 min

step 1 find len

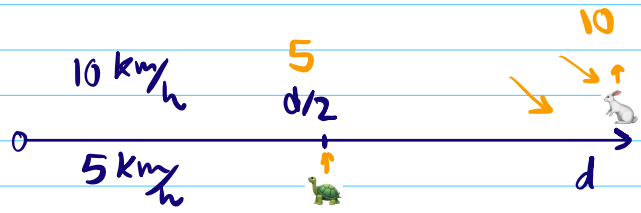
```
len = 0
cur = head
while(cur != null){
    len++;
    cur = cur.next;
}
```
O(n)

step 2 & go len/2 from head

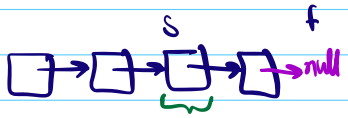
```
cur = head
for(i = 1; i <= len/2; i++)
    cur = cur.next
}
ret cur
```
O(n/2) ~ O(n)

TC & O(n) SC & O(1)

idea 2



fast pointer_
slow pointer
for Linked List



slow fast

Code

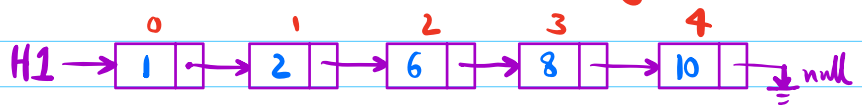
```
s = head   f = head
while ( f != null && f.next != null ){
    s = s.next
    f = f.next.next
}
ret s
```

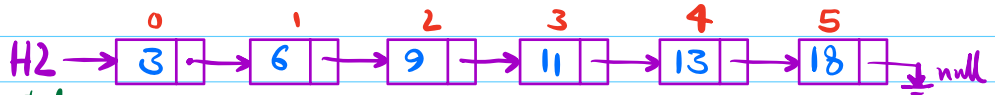
? Also what about slow

TC: O(n/2) ~ O(n) SC: O(1)

P2 Merge two sorted linked List into a single sorted list.

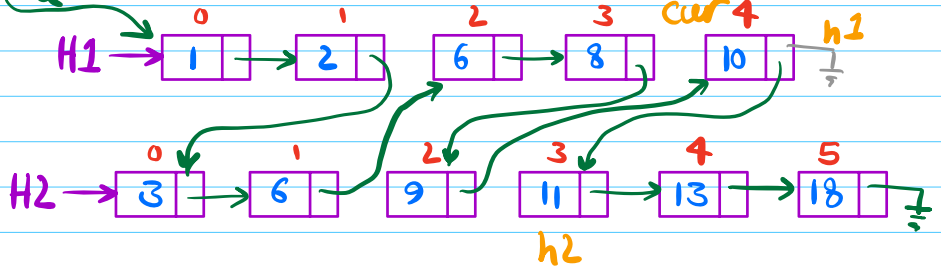
Constraint: Do not create new list, only update pointers





3 min

sorted Head: 1 → 2 → 3 → 6 → 6 → 8 → 9 → 10 → 11 → 13 → 18



Quiz

TC: O(m+n)
SC: O(1)
vs Arr SC: O(m+n)

h1 == H1
h2 == H

```
if(h1 == null) ret h2
if(h2 == null) ret h1
if (h1.data < h2.data){
    head = h1
    h1 = h1.next
}else{
    head = h2
    h2 = h2.next
}
```

set head

var head is for merged list

```
cur = head
while(H1 != null && H2 != null){
    if( H1.data <= H2.data){
        cur.next = H1
        H1 = H1.next
    }else
        cur.next = H2
        H2 = H2.next
    }
    cur = cur.next
}
if(H1 != null) cur.next = h1
if(H2 != null) cur.next = H2
ret head
```

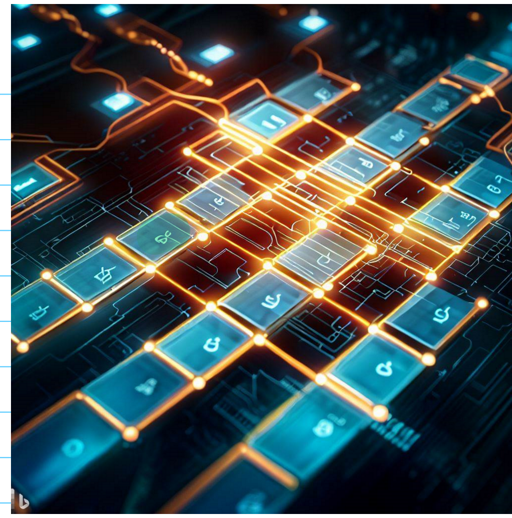
m + n

main logic

which list has more items

Generated by Bing AI image creator
https://www.bing.com/images/create

Query: “merge sort array linked list algorithm”



P3 Sort a given linked list using merge sort.

Constraint: Do not create new list, only update pointers

ex H → 1 → 4 → 3 → 6 → 5 → 2 → null

(indices 0, 1, 2, 3, 4, 5; h2 at 0, h2 at 3)

ret H $\xrightarrow{ans}$ 1 → 2 → 3 → 4 → 5 → 6

hint

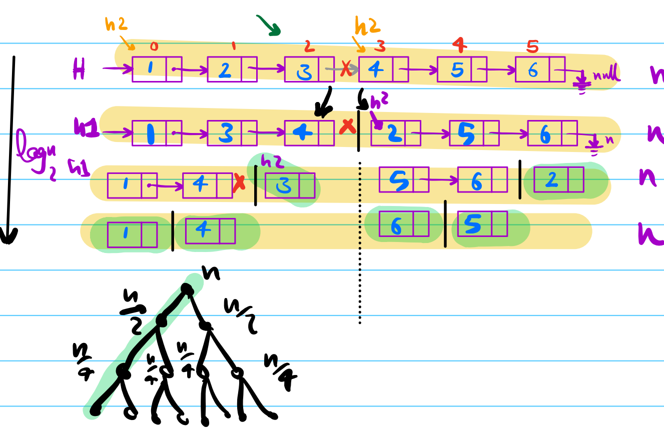
2 min

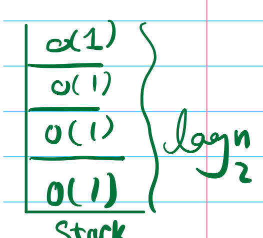
(P1) mid H ← getMid(H)   (P2) H ← merg(H1, H2)

Quiz
TC: O(nlogn)
SC: O(logn)

vs
Arr
TC: O(nlogn)
SC: O(n)

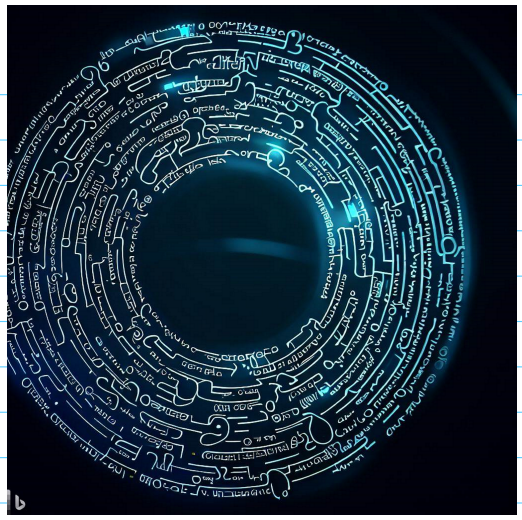
```
node meregSort(head) {
    if (head == null || head.next == null) ret head
        // len=0          len=1
    h = head
    node mid = getMid(h)      O(n)
    h1 = h;   h2 = mid.next       yes needed
    mid.next = null  → split (do we need this?)   cut it
    h1 = meregSort(h1);
    h2 = meregSort(h2);
    ret merg(h1, h2)
}
```



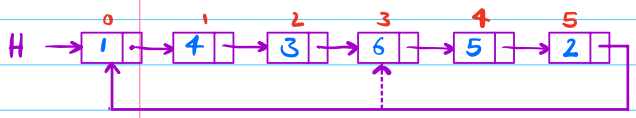


Generated by Bing AI image creator
https://www.bing.com/images/create

Query: "circular linked list code algorithm visualization"



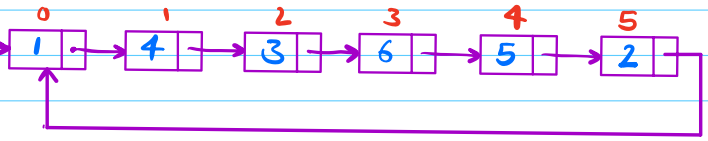
circle linked list



true/false

P4 Check if a given linked list has a cycle.

2 min ex1? H



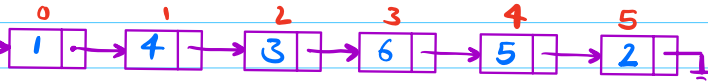
→ true

hashset<Node>

↓

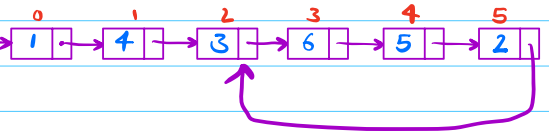
SC=O(n)

ex2? H

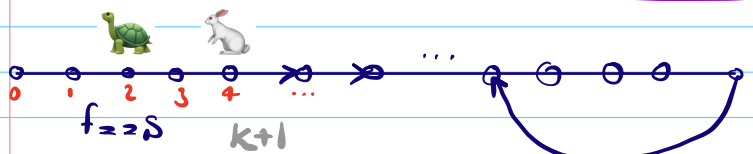


→ null → false

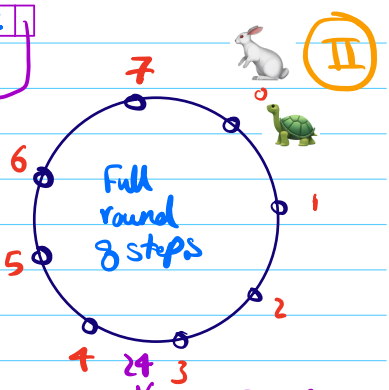
ex3



I



II

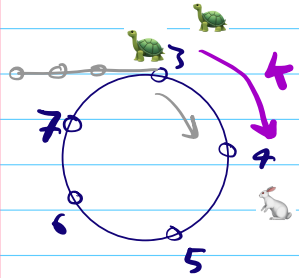


statement

if cycle ⇒ f and s will meet on a node/why?

math Proof

beyond scope



3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11

n

n≥ size of cycle

10

k≥1

max after n → steps

① k+2-1 = k+1

② (k+1)+2-1 = k+2

③ (k+2)+2-1 = k+3

⋮

(n) ... = k+n

d = 8 { same node

d = size of circle

k
k+1
k+2
k+3
⋮
k+n

worst case: takes size of cycle to meet

Code

```
s = head ; f = head
while (f != null && f.next != null){
    s = s.next
    f = f.next.next
    if (s == f) ret true
}
ret false
```

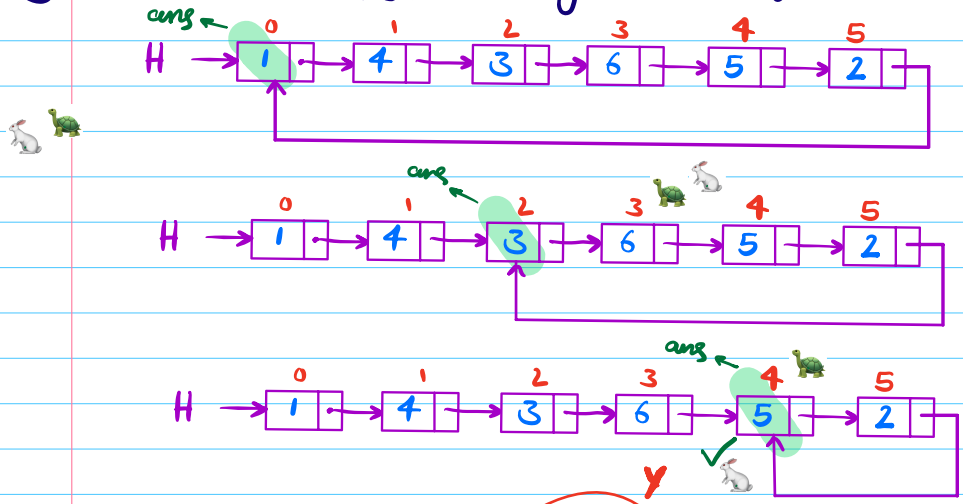
pointer compare

TC: O(n)

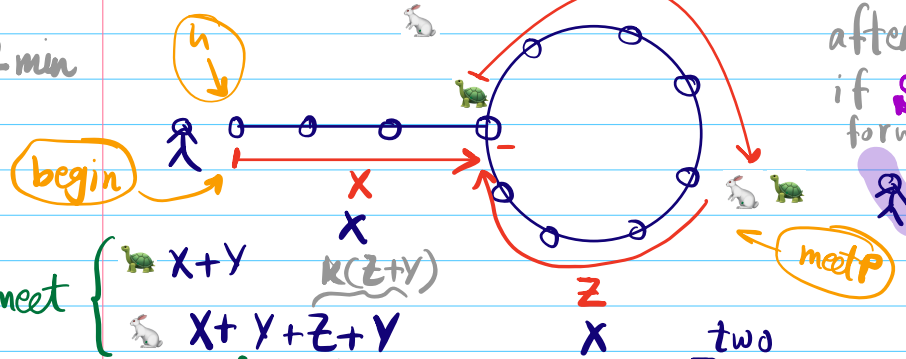
SC: O(1)

* domain knowledge; you are not expected to infere all in interview

#P5 Given a linked list with cycle, find the start of cycle



2 min



after T unit of time if S went 3 steps (dots) forward, how many steps did f go?

2S = 1f

two

$(X+Y)\,2 = X+Y+Z+Y$ at meeting point

$X+X+X+X = X+X+Z+Y$

$X = Z$

$$X = Z + \frac{(k-1)(Z+Y)}{C1}$$

Code

```
s = head ; f = head
while (f != null && f.next != null){
    s = s.next
    f = f.next.next
    if (s == f) break ✓ they met!
}
meetP = s      | x point forward to meet
begin = h      | the start of cycle

while (meetP != begin){
    meetP = meetP.next
    begin = begin.next
}
ret meetP → definitly start of cycle!
```

O(n) TC

O(1) SC