143. Reorder List

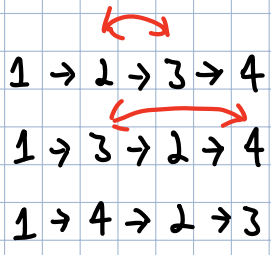
You are given the head of a singly linked list.

Re-order the list like so:

$L_0 \to L_1 \to \dots \to L_{n-1} \to L_n \Rightarrow L_0 \to L_n \to L_1 \to L_{n-1} \to L_2 \to L_{n-2} \to \dots$

Ex.1

Input: [1 2 3 4]
Output: [1 4 2 3]



1 2 3 4 5 6 7 8 1 8 2 7 3 6 4 5

1 3 2 4 5 6 7 8
1 4 2 3 5 6 7 8
1 5 2 3 4 6 7 8
1 6 2 3 4 5 7 8
1 7 2 3 4 5 6 8
1 8 2 3 4 5 6 7 X

1 → 2 → 3 → 4 → 5 → 6 → \0

1 6 2

1 → 6 → 2 → 5 →

1 → 2 → 3 6 → 5 → 4

1 → 6 → 2 → 5 → 3 → 4

front → next = back → next

back → next → next = front → next → next

1 6 2 5 3 4

tmp1 = nullptr
tmp2 = nullptr

while (rhs && lhs)
  tmp1 = rhs → next
  tmp2 = lhs → next
  lhs → next = rhs

1) Find mid-point of linked list
2) Reverse rhs of linked list
3) lhs & rhs, insert rhs into lhs

```
tmp = lhs → next
lhs → next = rhs
tmp1 = rhs → next
rhs → next = tmp
lhs = tmp
rhs = tmp1
```

7.

1) How to find mid point?

```
slow = head   // Assume `head` is not nullptr
fast = head
while   fast → next and fast → next → next
        slow = slow → next
        fast = fast → next → next
```

2.) How to reverse?

1 → 2 → 3 → 4 => 4 → 3 → 2 → 1

prev = nullptr
current = head
next = nullptr

while current
    next = current->next
    current->next = prev
    prev = current
    current = next

return prev

3.) How to insert?

1 → 2 → 3 → nullptr
6 → 5 → 4 → nullptr

1 → 6 → 2 → 5 → 3 → 4 → nullptr

1 2 3
↓
6 → 5 4

lhs1 = null
rhs1 = null