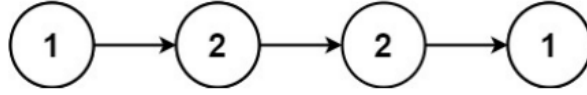


Palindrome Linked List

Given the head of a singly linked list, return true if it is a palindrome or false otherwise.

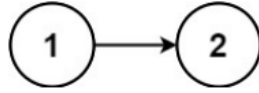
Example 1:



Input: head = [1,2,2,1]

Output: true

Example 2:



Input: head = [1,2]

Output: false

Constraints:

- The number of nodes in the list is in the range [1, 10⁵].
- 0 ≤ Node.val ≤ 9

الحل لهذه المشكلة يعتمد على التحقق مما إذا كانت قائمة مرتبطة مفردة تمثل متسلسلة من الأرقام التي تكون نفسها عند قراءتها من البداية أو النهاية (مثلما يحدث مع الكلمات أو الأرقام التي تُعد "متناظرة").

خطة الحل:

١. إيجاد منتصف القائمة:

يمكن استخدام أسلوب "العدائين" (Fast and Slow Pointer)، حيث يتحرك مؤشر بطيء بمقدار خطوة واحدة، بينما يتحرك مؤشر سريع بمقدار خطوتين. عندما يصل المؤشر السريع إلى نهاية القائمة، يكون المؤشر البطيء في منتصف القائمة.

٢. عكس النصف الثاني من القائمة:

بعد إيجاد منتصف القائمة، نعكس النصف الثاني منها.

٣. مقارنة النصفين:

نبدأ من بداية القائمة ومن منتصف القائمة بعد عكسها ونتحقق مما إذا كانت القيم متماثلة. إذا كانت جميع القيم متماثلة، فالقائمة متناظرة (Palindrome).

٤. إعادة القائمة إلى حالتها الأصلية:

يمكن إعادة النصف المعكوس إلى ترتيبه الأصلي بعد الانتهاء من المقارنة إذا لزم الأمر.

الشرح البرمجي:

```
:class ListNode
:
:   def __init__(self, val=0, next=None)
:       self.val = val
:       self.next = next
:
:
:   def isPalindrome(self, head: ListNode) -> bool
:       if not head or not head.next
:           return True
```

١. إيجاد منتصف القائمة

slow = head

fast = head

:while fast and fast.next

slow = slow.next

fast = fast.next.next

٢. عكس النصف الثاني من القائمة

prev = None

:while slow

next_node = slow.next

slow.next = prev

prev = slow

slow = next_node

٣. مقارنة النصف الأول بالنصف الثاني المعكوس

left = head

right = prev

:while right # نتحقق فقط من النصف الثاني

:if left.val != right.val

return False

left = left.next

right = right.next

٤. إذا كان كل شيء متماثل، فالقائمة متناظرة

return True

الشرح:

١. قمنا بإيجاد منتصف القائمة باستخدام طريقة المؤشرين.

٢. قمنا بعكس النصف الثاني من القائمة.

٣. قمنا بمقارنة قيم النصف الأول بالنصف الثاني المعكوس.

٤. إذا كانت جميع القيم متساوية، نعيد True، وإذا وجدنا أي اختلاف نعيد False.