

# 876. Middle of the Linked List

☰ Tags	
📅 Property	@September 12, 2022

## Question

原文：

Given the `head` of a singly linked list, return *the middle node of the linked list*.

If there are two middle nodes, return **the second middle** node.

我的理解：

給定一串鏈結陣列，返回中間的node的數值，如果node個數是偶數個有返回後面那個node數值

翻譯：

給出一個單鏈表的"頭"，返回鏈表的中間節點。

如果有兩個中間節點，返回**第二個**中間節點。

自評翻譯正確性：70% (要返回的是包含該node與其後方的整串)

- Word Memory：

## Code

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
```

```

public:
    ListNode* middleNode(ListNode* head) {
        ListNode * pre=head;
        ListNode * pro=head;
        int i=0,j=0;
        if(head->next==NULL){
            return head;
        }
        while(pre!=NULL){
            pre=pre->next;
            i++;
        }

        i=i/2;

        while(j<i){
            pro=pro->next;
            j++;
        }
        return pro;
    }
};

```

思路：先開一個 pre 去遍歷抓整個list長度，紀錄為i，然後i除2，之後再用一個 pro 遍歷並用 j 記步，一直到j==i就是一半的位置

**Success** [Details >](#)

Runtime: **0 ms**, faster than **100.00%** of C++ online submissions for Middle of the Linked List.

Memory Usage: **7.1 MB**, less than **71.98%** of C++ online submissions for Middle of the Linked List.

Next challenges:

[Delete the Middle Node of a Linked List](#)

[Maximum Twin Sum of a Linked List](#)

Show off your acceptance:



Time Submitted	Status	Runtime	Memory	Language
09/11/2022 22:51	Accepted	0 ms	7.1 MB	cpp

## 優良code參考

```
class Solution {
public:
    ListNode* middleNode(ListNode* head) {
        ListNode *slow = head, *fast = head;
        while (fast && fast->next) {
            slow = slow->next;
            fast = fast->next->next;
        }
        return slow;
    }
};
```

思路：分成兩個pointer下去跑，slow & fast，fast跑的是slow的兩倍快，所以如果fast已經抵達終點或者說已經不能再往下跑了停止時，slow跑的剛好是fast的一半，也就剛好是在一半的位置。  
(great solution)