**234. Palindrome Linked List**

Easy

1961283FavoriteShare

Given a singly linked list, determine if it is a palindrome.

**Example 1:**

**Input:** 1->2

**Output:** false

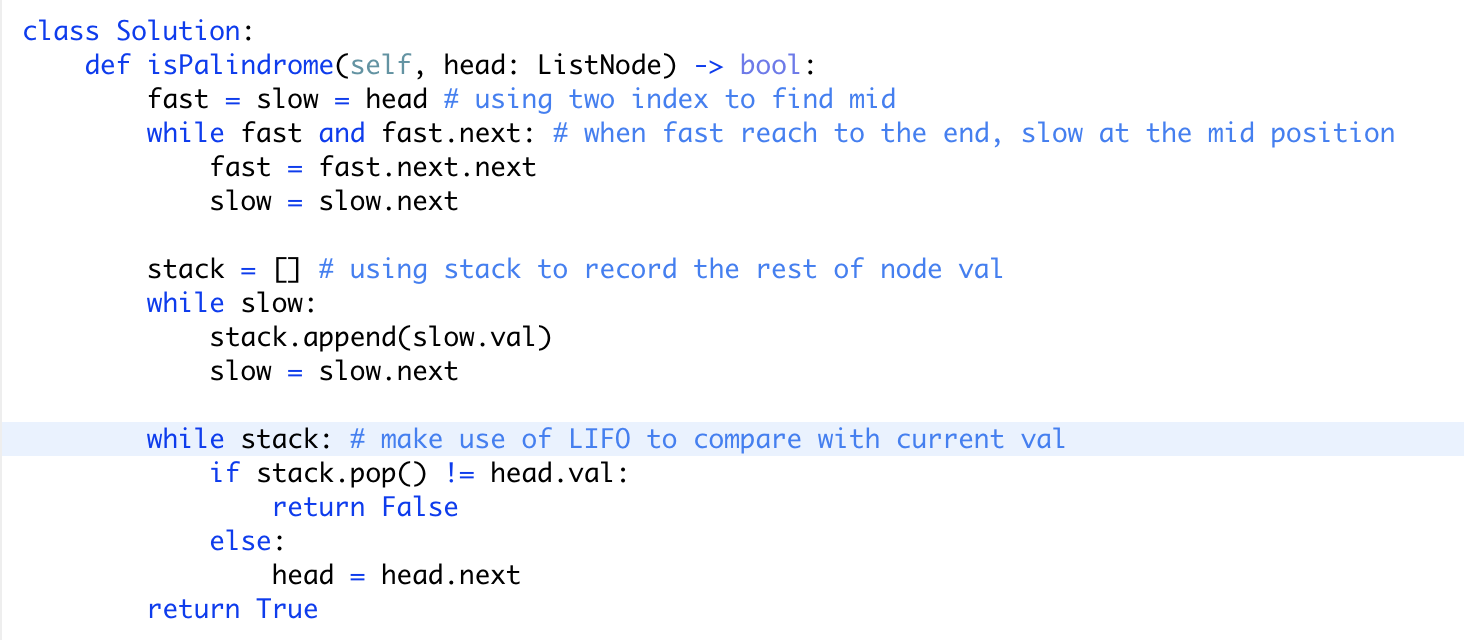
**Example 2:**

**Input:** 1->2->2->1

**Output:** true

**Follow up:**  
Could you do it in O(n) time and O(1) space?

1. Stack



Time: O(n)

Space: O(n)

以及理论上此题空间复杂度不可能是O(1)

It is a common misunderstanding that the space complexity of a program is just how much the size of additional memory space being used besides input. An important prerequisite is neglected the above definition: [the input has to be read-only](https://en.wikipedia.org/wiki/DSPACE#Machine_models). By definition, changing the input and change it back is not allowed (or the input size should be counted when doing so). Another way of determining the space complexity of a program is to simply look at how much space it has written to. Reversing a singly linked list requires writing to O(n) memory space, thus the space complexities for all "reverse-the-list"-based approaches are O(n), not O(1).

Solving this problem in O(1) space is theoretically impossible due to two simple facts: (1) a program using O(1) space is computationally equivalent to a finite automata, or a regular expression checker; (2) [the pumping lemma](https://en.wikipedia.org/wiki/Pumping_lemma_for_regular_languages) states that the set of palindrome strings does not form a regular set.

Please change the incorrect problem statement.

简单来说，只有当输入是只读的时候，空间复杂度才不受输入大小的影响。所以任何基于对输入修改的算法，比如翻转列表等，都是需要O(n)的时间复杂度。