**141. Linked List Cycle**

Easy

1657192FavoriteShare

Given a linked list, determine if it has a cycle in it.

To represent a cycle in the given linked list, we use an integer pos which represents the position (0-indexed) in the linked list where tail connects to. If pos is -1, then there is no cycle in the linked list.

**Example 1:**

**Input:** head = [3,2,0,-4], pos = 1

**Output:** true

**Explanation:** There is a cycle in the linked list, where tail connects to the second node.



**Example 2:**

**Input:** head = [1,2], pos = 0

**Output:** true

**Explanation:** There is a cycle in the linked list, where tail connects to the first node.



**Example 3:**

**Input:** head = [1], pos = -1

**Output:** false

**Explanation:** There is no cycle in the linked list.



**Follow up:**

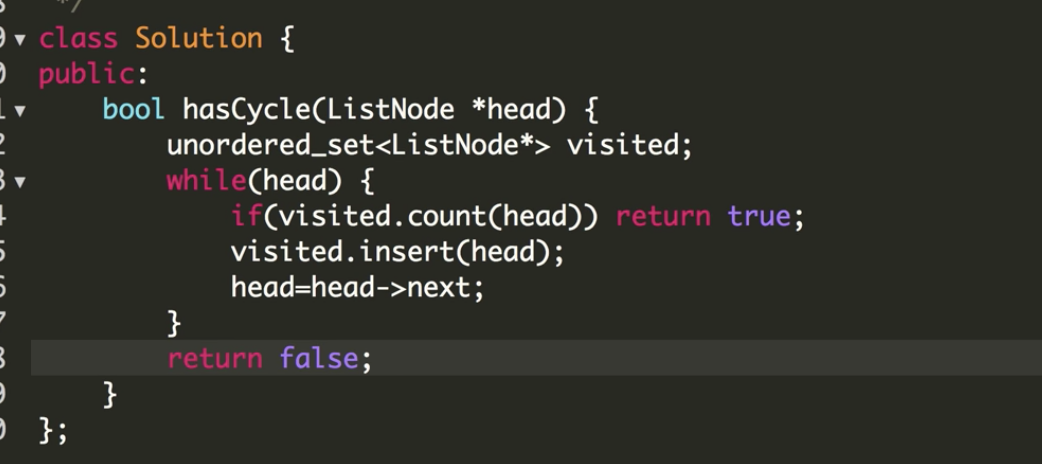
Can you solve it using *O(1)* (i.e. constant) memory?

读清题目，pos只用来构造list，不会真的出现在输入里。

1. Hash set

C++

<https://blog.csdn.net/cumirror/article/details/5596908>



关于python 哈希表

<https://python-data-structures-and-algorithms.readthedocs.io/zh/latest/07_%E5%93%88%E5%B8%8C%E8%A1%A8/hashtable/> 这个文章写的特别好