

LEETCODE BOOTCAMP

ASSIGNMENT 3

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def isPalindrome(self, head: Optional[ListNode]) -> bool:
        slow, fast = head, head
        while fast and fast.next:
            slow = slow.next
            fast = fast.next.next
        prev, curr = None, slow
        while curr:
            nxt = curr.next
            curr.next = prev
            prev = curr
            curr = nxt
        while prev:
            if prev.val != head.val: return False
            prev, head = prev.next, head.next
        return True
```

Problem List < > > Run Submit

Description Editorial Solutions Accepted Submissions

All Submissions

Accepted 93 / 93 testcases passed
XoITACHIXo submitted at Mar 19, 2025 01:10

Editorial Solution

Runtime 38 ms | Beats 43.54%
Memory 34.70 MB | Beats 92.88%

Analyze Complexity

Code Python3

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def isPalindrome(self, head: Optional[ListNode]) -> bool:
        slow, fast = head, head
```

More challenges

206. Reverse Linked List 1120. Maximum Twin Sum of a Linked List

Code

```
Python3 Auto
1 # Definition for singly-linked list.
2 # class ListNode:
3 #     def __init__(self, val=0, next=None):
4 #         self.val = val
5 #         self.next = next
6 class Solution:
7     def isPalindrome(self, head: Optional[ListNode]) -> bool:
8         slow, fast = head, head
9         while fast and fast.next:
10             slow = slow.next
11             fast = fast.next.next
12         prev, curr = None, slow
13         while curr:
14             nxt = curr.next
15             curr.next = prev
```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

head = [1,2,2,1]

Output

true

Expected

true

Contribute a testcase

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def isPalindrome(self, head: Optional[ListNode]) -> bool:
        slow, fast = head, head
        while fast and fast.next:
            slow = slow.next
            fast = fast.next.next
        prev, curr = None, slow
        while curr:
            nxt = curr.next
            curr.next = prev
            prev = curr
            curr = nxt
        while prev:
            if prev.val != head.val: return False
            prev, head = prev.next, head.next
        return True
```

Problem List < > ✕

Description Editorial Solutions Accepted ✕ Submissions

← All Submissions

Accepted 12 / 12 testcases passed
XoITACHIxo submitted at Mar 19, 2025 01:12

Editorial Solution

Runtime 1 ms | Beat: 60.45%
Memory 23.31 MB | Beats 38.27%

Analyze Complexity

Code Python3

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def reorderList(self, head: Optional[ListNode]) -> None:
```

View more

More challenges

2095. Delete the Middle Node of a Linked List

Python3 Auto

```
prev, curr = None, slow.next
while curr:
    nextt = curr.next
    curr.next = prev
    prev = curr
    curr = nextt
slow.next = None

# Step 3: merge lists
head1, head2 = head, prev
while head2:
    nextt = head1.next
    head1.next = head2
    head1 = nextt
    head2 = nextt
```

Saved Ln 33, Col 26

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

head =
[1,2,3,4]

Output

[1,4,2,3]

Expected

[1,4,2,3]

Contribute a testcase

```
class Solution:
    def setZeroes(self, matrix: List[List[int]]) -> None:
        """
        Do not return anything, modify matrix in-place instead.
        """
        m, n = len(matrix), len(matrix[0])
        firstRowZero = any(matrix[0][j] == 0 for j in range(n))
        firstColZero = any(matrix[i][0] == 0 for i in range(m))

        # Mark zeros in the first row and column
        for i in range(1, m):
            for j in range(1, n):
                if matrix[i][j] == 0:
                    matrix[i][0] = 0
                    matrix[0][j] = 0

        # Set matrix cells to zero based on markers
        for i in range(1, m):
```

```

        for j in range(1, n):
            if matrix[i][0] == 0 or matrix[0][j] == 0:
                matrix[i][j] = 0

# Zero out the first row if needed
if firstRowZero:
    for j in range(n):
        matrix[0][j] = 0

# Zero out the first column if needed
if firstColZero:
    for i in range(m):
        matrix[i][0] = 0

```

Problem List < >

Description Editorial Solutions Accepted Submissions

All Submissions

Accepted 202 / 202 testcases passed

✶ **koitachiko** submitted at Mar 19, 2025 01:14

Editorial Solution

Runtime

1 ms Beats 81.32%

Analyze Complexity

Memory

18.60 MB Beats 19.17%

40%

20%

0%

0ms 2ms 4ms 6ms 8ms 10ms 12ms 14ms

Code | Python3

```

class Solution:
    def setZeroes(self, matrix: List[List[int]]) -> None:
        """
        Do not return anything, modify matrix in-place instead.
        """
        m, n = len(matrix), len(matrix[0])
        firstRowZero = any(matrix[0][j] == 0 for j in range(n))
        firstColZero = any(matrix[i][0] == 0 for i in range(m))

        for i in range(1, m):
            for j in range(1, n):
                if matrix[i][0] == 0 or matrix[0][j] == 0:
                    matrix[i][j] = 0

        # Zero out the first row if needed
        if firstRowZero:
            for j in range(n):
                matrix[0][j] = 0

        # Zero out the first column if needed
        if firstColZero:
            for i in range(m):
                matrix[i][0] = 0

```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

matrix =

[[1,1,1],[1,0,1],[1,1,1]]

Output

[[1,0,1],[0,0,0],[1,0,1]]

Expected

[[1,0,1],[0,0,0],[1,0,1]]

01:14 19-03-2025