Problem | Editorial | Submissions | Comments

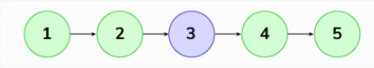
Middle of a Linked List

Difficulty: **Easy** Accuracy: **57.93%** Submissions: **336K+** Points: **2**

Given the head of a linked list, the task is to find the **middle**. For example, the middle of **1->2->3->4->5** is **3**. If there are two middle nodes (even count), return the **second middle**. For example, middle of **1->2->3->4->5->6** is **4**.

Examples:

Input: Linked list: 1->2->3->4->5
Output: 3



Explanation: The given linked list is 1->2->3->4->5 and its middle is 3.

Input: Linked list: 2->4->6->7->5->1
Output: 7

Explanation: The given linked list is 2->4->6->7->5->1 and its middle is 7.

Expected Time Complexity: O(n)
Expected Auxiliary Space: O(1)

Constraints:
1 <= no. of nodes <= 10^5

Try more examples

Seen this question in a real interview before ? Yes No

Company Tags

VMWare Zoho Flipkart Morgan Stanley Amazon Microsoft Samsung Hike Payu MAQ Software Adobe Wipro SAP Labs Qualcomm Nagarro GE Veritas IgniteWorld Netskope

Topic Tags

Related Interview Experiences

Related Articles

Report An Issue
If you are facing any issue on this page. Please let us know.

Java (1.8) Average Time: 20m Start Timer

```java
//{ Driver Code Starts
import java.io.*;
import java.util.*;

class Node {
    int data;
    Node next;

    Node(int x) {
        data = x;
        next = null;
    }
}

class GFG {
    static void printList(Node node) {
        while (node != null) {
            System.out.print(node.data + " ");
            node = node.next;
        }
        System.out.println();
    }

    public static void main(String args[]) throws IOException {
        BufferedReader read = new BufferedReader(new InputStreamReader(System.in));
        int t = Integer.parseInt(read.readLine());
        while (t > 0) {
            String str[] = read.readLine().trim().split(" ");
            int n = str.length;
            Node head = new Node(Integer.parseInt(str[0]));
            Node tail = head;
            for (int i = 1; i < n; i++) {
                tail.next = new Node(Integer.parseInt(str[i]));
                tail = tail.next;
            }
```

Custom Input Compile & Run Submit