

Convert Binary Number in a Linked List to Integer(December 7 2021 Tuesday)

Tuesday, December 7, 2021 9:11 AM

1290. Convert Binary Number in a Linked List to Integer

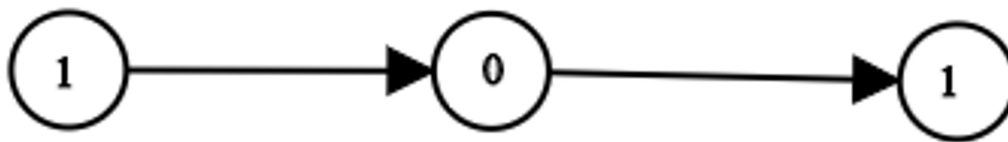
Easy

179786Add to ListShare

Given `head` which is a reference node to a singly-linked list. The value of each node in the linked list is either 0 or 1. The linked list holds the binary representation of a number.

Return the *decimal value* of the number in the linked list.

Example 1:



Input: head = [1,0,1]

Output: 5

Explanation: (101) in base 2 = (5) in base 10

Example 2:

Input: head = [0]

Output: 0

Example 3:

Input: head = [1]

Output: 1

Example 4:

Input: head = [1,0,0,1,0,0,1,1,1,0,0,0,0,0,0]

Output: 18880

Example 5:

Input: head = [0,0]

Output: 0

Constraints:

- The Linked List is not empty.
- Number of nodes will not exceed 30.
- Each node's value is either 0 or 1.

From <<https://leetcode.com/problems/convert-binary-number-in-a-linked-list-to-integer/>>

NAÏVE APPROACH !!

```
/**
 * Definition for singly-linked list.
 *
 * public class ListNode {
 *
 *     int val;
 *
 *     ListNode next;
 *
 *     ListNode() {}
 *
 *     ListNode(int val) { this.val = val; }
 *
 *     ListNode(int val, ListNode next) { this.val = val; this.next = next; }
 *
 * }
 */
class Solution {
    public int getDecimalValue(ListNode head) {
        ListNode temp= head;
        String ans="";

        while(temp!=null)
        {
            if(temp.next==null)
                System.out.print(temp.val);
            else
                System.out.print(temp.val+"-->");
            ans+=temp.val;

            temp=temp.next;
        }
        int value =0;
        int n=ans.length();

        for(int i=n-1;i>=0;i--)
        {
            value+=(Integer.parseInt(""+ans.charAt(i)))*Math.pow(2,(n-1)-i);
        }
        return value;
    }
}
```

CODE PART

```
class Solution {

    public int getDecimalValue(ListNode head) {

        ListNode temp= head;

        String ans="";

        while(temp!=null)

        {

            if(temp.next==null)
                System.out.print(temp.val);

            else
                System.out.print(temp.val+"--->");

            ans+=""+temp.val;

            temp=temp.next;

        }

        int value =0;

        int n=ans.length();

        for(int i=n-1;i>=0;i--)
        {
            value+=(Integer.parseInt( ""+ ans.charAt(i))) * Math.pow(2,(n-1)-i);
        }

        return value;

    }
}
```

OPTIMIZED APPROACH !!!

```

class Solution {
    public int getDecimalValue(ListNode head) {

        ListNode node = head;
        int decimalNumber = 0;

        while(node != null){

            decimalNumber = 2 * decimalNumber + node.val;
            node = node.next;
        }

        return decimalNumber;

    }
}

```

CODE PART

```

class Solution {
    public int getDecimalValue(ListNode head) {

        ListNode node = head;
        int decimalNumber = 0;

        while(node != null){

            decimalNumber = 2 * decimalNumber + node.val;
            node = node.next;
        }

        return decimalNumber;

    }
}

```