Convert binary number in a linked list to Integer

LeetCode Explore Problems Interview New Contest Discuss Store LeetCode is hiring! Apply NOW

Description Solution Discuss (999+) Submissions

1290. Convert Binary Number in a Linked List to Integer

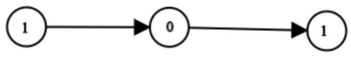
Easy 3122 140 Add to List Share

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.

The **most significant bit** is at the head of 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
```

First we need to see How to convert Binary to decimal

let 1 0 1 0

$0 \times 2^0 = 0$

$1 \times 2^1 = 2$

$0 \times 2^2 = 0$

$1 \times 2^3 = 8$

$= 10$

1 → 0 → 1

← since we read From Left to Right

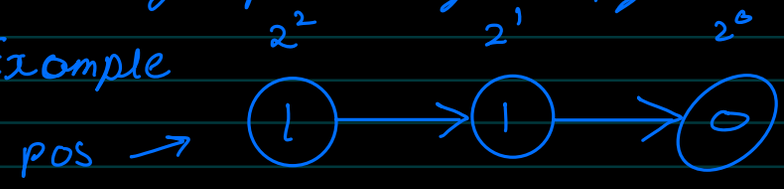
First approach :- we need to reserve the linked list and simply use our formula to convert into decimal

Formula

```
int ans = 0, i = 0;
while(temp != Null)
ans = ans + (Data * 2**i)   (Basic formula)
i++;
```

Second approach:- we will find the length of LL then so without reversing the linked list we can get power by simply (Len - Pos)

Example

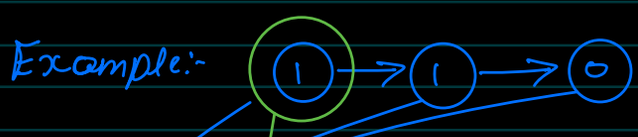


len = 3
pos = 1

TC :- O(n)
SC :- O(1)

```
while(temp != Null){
    Ans = Ans + Data * 2**(len-pos)
    pos++
}
```

Third approch: For above approch we need to know the length of LL so to avoid that we will consider our first node as First bit and we will check if it our first bit or not if not the we will multiple previous Answer with 2




```
int ans=0
while(temp != Null)
{
    (ans = ans * 2)
    ans = ans + temp->data
}
```

Consider this node as Ans

we say it our first bit
i.e 2^0

ans = 0 * 2 = 0

ans = 0 + 1 (temp->data)
ans = 1

Checked if we ended LL

Ans = 3 * 2 (Previous * 2)
= 6

Ans = 6 + temp->data(0)

Ans = 6

Checked if we ended LL

so Ans = 1 * 2 (Previous * 2)

Ans = 2 + temp->data(1)
= 3

we ended LL so we got the answer 6

Optimal code

```
class Solution {
public:
    int getDecimalValue(ListNode* head) {
        int ans=0;
        ListNode* temp=head;
        while(temp!=NULL){
            // ans=ans*2;
            ans=ans*2+temp->val;
            temp=temp->next;
        }
        return ans;
    }
};
```

T.C - O(n)

S.C - O(1)