382. Linked List Random Node

QuestionEditorial Solution

My Submissions

- Total Accepted: **4856**
- Total Submissions: **10456**
- Difficulty: **Medium**

Given a singly linked list, return a random node's value from the linked list. Each node must have the **same probability** of being chosen.

Follow up:

What if the linked list is extremely large and its length is unknown to you? Could you solve this efficiently without using extra space?

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    /** @param head The linked list's head.
        Note that the head is guaranteed to be not null, so it contains at least one node.
*/
    ListNode *head;
    Solution(ListNode* head) {
        this->head = head;
    }

    /** Returns a random node's value. */
    int getRandom() {
        int res = head->val;
        ListNode *node = head->next;
        int i=2;
        while(node)
        {
            int j=rand()%i;
            if(j==0)
            {
                res = node->val;
            }
            i++;
            node = node->next;
        }
        return res;
    }
};
```

```
/**
 * Your Solution object will be instantiated and called as such:
 * Solution obj = new Solution(head);
 * int param_1 = obj.getRandom();
 */
```