Problem 382: Linked List Random Node

Problem Information

Difficulty: Medium
Acceptance Rate: 64.44%
Paid Only: No
Tags: Linked List, Math, Reservoir Sampling, Randomized

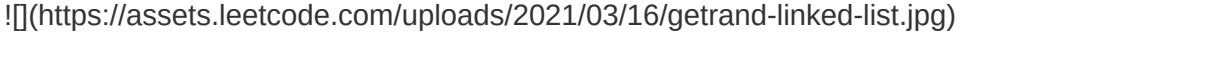
Problem Description

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.

Implement the `Solution` class:

* `Solution(ListNode head)` Initializes the object with the head of the singly-linked list `head`. * `int getRandom()` Chooses a node randomly from the list and returns its value. All the nodes of the list should be equally likely to be chosen.

Example 1:



Input ["Solution", "getRandom", "getRandom", "getRandom", "getRandom", "getRandom"] [[[1, 2, 3]], [], [], [], [], []] **Output** [null, 1, 3, 2, 2, 3] **Explanation** Solution solution = new Solution([1, 2, 3]); solution.getRandom(); // return 1 solution.getRandom(); // return 3 solution.getRandom(); // return 2 solution.getRandom(); // return 2 solution.getRandom(); // return 3 // getRandom() should return either 1, 2, or 3 randomly. Each element should have equal probability of returning.

Constraints:

* The number of nodes in the linked list will be in the range `[1, 104]`. * `-104 <= Node.val <= 104` * At most `104` calls will be made to `getRandom`.

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?

Code Snippets

C++:

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
Solution(ListNode* head) {

}

int getRandom() {

}
};

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

Java:

```
/**
* 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 Solution(ListNode head) {

}

public int getRandom() {

}
}

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

Python3:

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

def __init__(self, head: Optional[ListNode]):


def getRandom(self) -> int:



# Your Solution object will be instantiated and called as such:
# obj = Solution(head)
# param_1 = obj.getRandom()
```