

水塘抽样

给你一个未知长度的链表，请你设计一个算法，只能遍历一次，随机地返回链表中的一个节点。

对随机抽样算法：当遇到第 i 个元素时，应该有 $\frac{1}{i}$ 的概率选在该元素， $1 - \frac{1}{i}$ 的概率不变。

```
class Solution {
public:
    Solution(ListNode *head) {
        _head = head;
    }

    int getRandom() {
        ListNode *p = _head;
        int i = 0;
        int res = 0;

        while (p) {
            if (rand() % (++i) == 0) { // 此处保证了为0的概率为1/i
                res = p->val;
            }

            p = p->next;
        }
        return res;
    }

private:
    ListNode *_head;
};
```

这类题目的难度在于证明算法随机的正确性：假设一共有 n 个元素，证明其随机性就是证明抽取每个元素的概率都为 $\frac{1}{n}$ 。

1. 对第 i 个元素，选中该元素的概率为 $\frac{1}{i}$;
2. 当增加一个元素时，选中该元素不变的概率为 $1 - \frac{1}{i+1}$;
3. 第 n 次，该元素不被替换的概率为 $\frac{1}{n}$ 。

所以选中第 i 个元素的概率为：
$$\frac{1}{i} * (1 - \frac{1}{i+1}) * (1 - \frac{1}{i+2}) * \dots * (1 - \frac{1}{n}) \rightarrow \frac{1}{i} * \frac{i}{i+1} * \dots * \frac{(n-1)}{n} \rightarrow \frac{1}{n}$$

对随机选取 k 个元素：要随机的选择 k 个元素，只要在第 i 个元素位置以 $\frac{k}{i}$ 的概率选择该元素，以 $1 - \frac{k}{i}$ 的概率保持不变。

```

// 随机返回链表的k个节点的值
std::vector<int> getRandom(ListNode *head, int k){
    std::vector<int> res(k,0);
    ListNode *p = head;

    // 前k个元素默认选上
    for(int i = k-1; i >= 0 && p;i++){
        res[i] = p->val;
        p = p->next;
    }

    int i = k;
    while(p) {
        int j = rand() % (++i);
        if(j < k) {
            res[j] = p->val;
        }
        p = p->next;
    }

    return res;
}

```

算法证明： $\frac{k}{i} * (1 - \frac{k}{i+1}) * \frac{1}{k} * \dots * (1 - \frac{k}{n}) * \frac{1}{k} \rightarrow \frac{k}{i} * (1 - \frac{1}{i+1}) * (1 - \frac{1}{i+2}) * \dots * (1 - \frac{1}{n}) \rightarrow \frac{k}{n}$