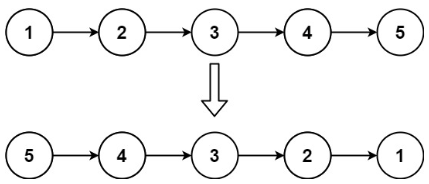
<https://leetcode.com/problems/reverse-linked-list>

**Reverse Linked List**

**Given the head of a singly linked list, reverse the list, and return the reversed list.**

Example:



Input: head = [1,2,3,4,5]

Output: [5,4,3,2,1]

Constraints:

The number of nodes in the list is the range [0, 5000].

-5000 <= Node.val <= 5000

**Method 1: (iteratively)**

Time Complexity: O(n) *[]*

Space Complexity: O(1) *[]*

ListNode\* reverseList(ListNode\* head) {

        ListNode \*ptr = head, \*tmp;

        head = NULL;

        while(ptr!=NULL){

            tmp = ptr;

            ptr= ptr->next;

            tmp->next = head;

            head = tmp;

        }

        return head;

    }

**Method 2: (recursively)**

int search(vector<int>& nums, int target) {

        int l = 0, r = nums.size()-1, mid;

        while(l<=r){

            mid = l + (r-l)/2;

            if(target==nums[mid])

                return mid;

            else if(target<nums[mid])

                r = mid - 1;

            else

                l = mid + 1;

        }

        return -1;

    }