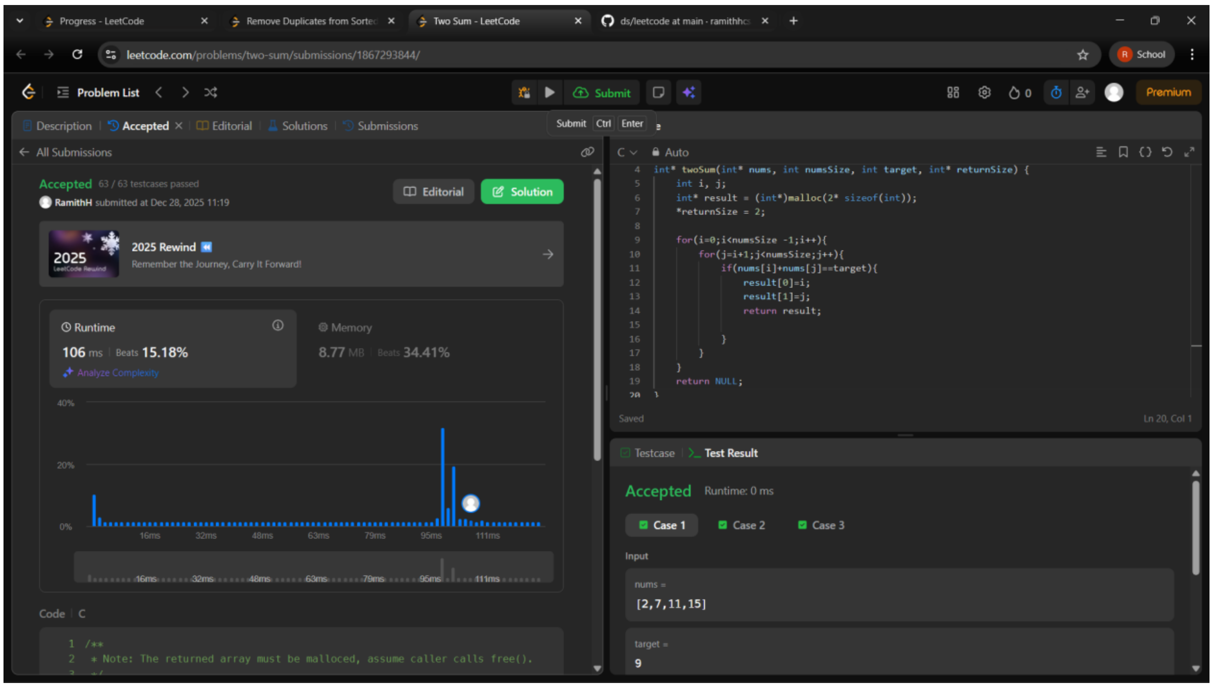
LEETCODE SOLUTIONS

1.TWO SUM



leetcode.com/problems/two-sum/submissions/1867293844/

Accepted 63 / 63 testcases passed

RamithH submitted at Dec 28, 2025 11:19

Runtime 106 ms Beats 15.18%

Memory 8.77 MB Beats 34.41%

Code | C

```c
/**
 * Note: The returned array must be malloced, assume caller calls free().
```

```c
int* twoSum(int* nums, int numsSize, int target, int* returnSize) {
    int i, j;
    int* result = (int*)malloc(2* sizeof(int));
    *returnSize = 2;

    for(i=0;i<numsSize -1;i++){
        for(j=i+1;j<numsSize;j++){
            if(nums[i]+nums[j]==target){
                result[0]=i;
                result[1]=j;
                return result;

            }
        }
    }
    return NULL;
}
```

Accepted Runtime: 0 ms

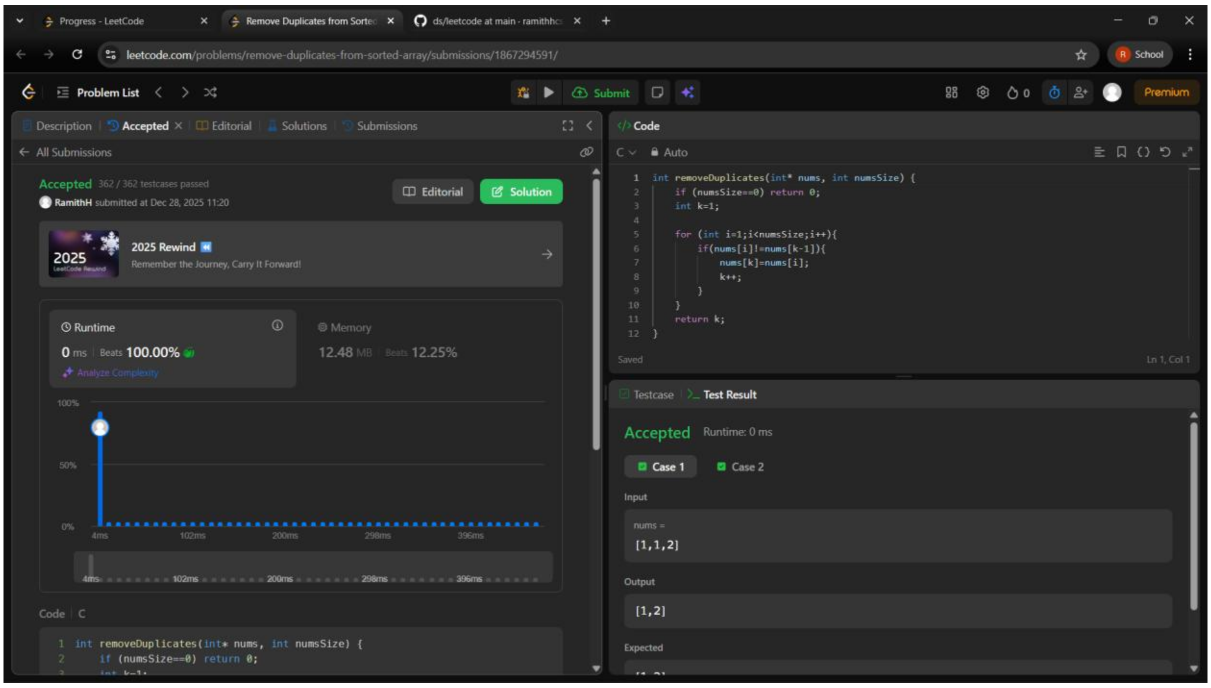
Case 1 Case 2 Case 3

Input

nums =
[2,7,11,15]

target =
9

26.Remove duplicates from sorted array



leetcode.com/problems/remove-duplicates-from-sorted-array/submissions/1867294591/

Accepted 362 / 362 testcases passed

RamithH submitted at Dec 28, 2025 11:20

Runtime 0 ms Beats 100.00%

Memory 12.48 MB Beats 12.25%

```c
int removeDuplicates(int* nums, int numsSize) {
    if (numsSize==0) return 0;
    int k=1;

    for (int i=1;i<numsSize;i++){
        if(nums[i]!=nums[k-1]){
            nums[k]=nums[i];
            k++;
        }
    }
    return k;
}
```

Accepted Runtime: 0 ms

Case 1 Case 2

Input

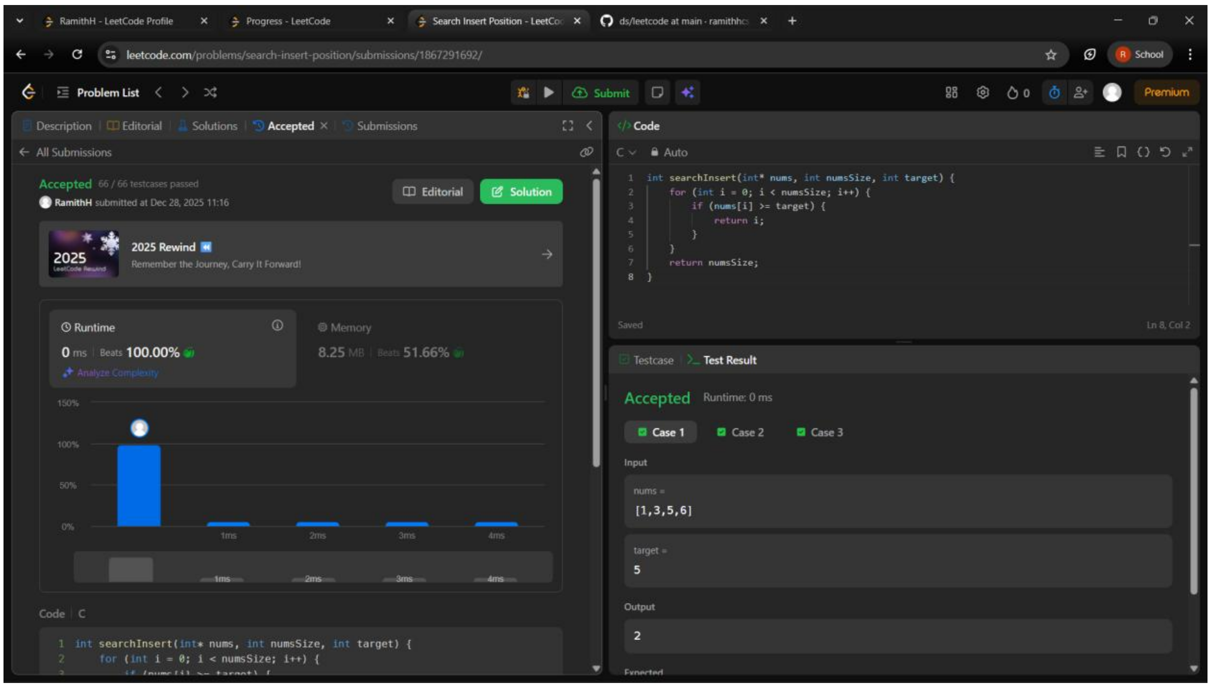
nums =
[1,1,2]

Output

[1,2]

Expected

35.Search insert position



Accepted 66 / 66 testcases passed

RamithH submitted at Dec 28, 2025 11:16

Runtime 0 ms | Beats 100.00%

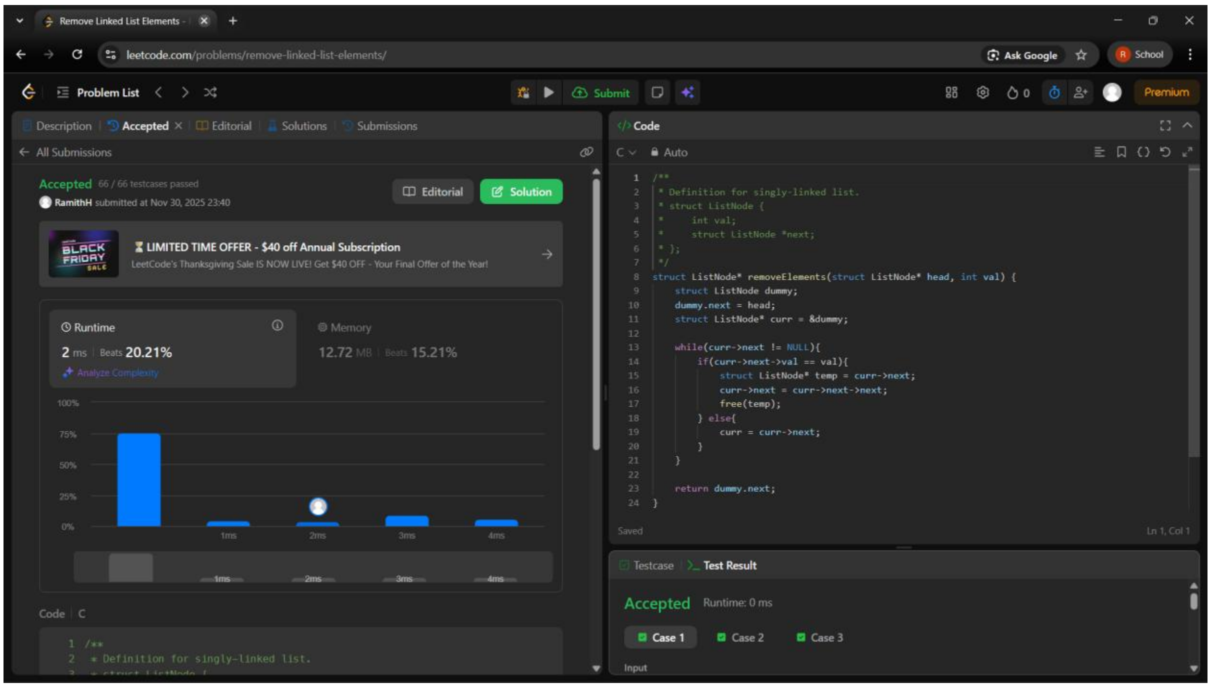
Memory 8.25 MB | Beats 51.66%

```c
int searchInsert(int* nums, int numsSize, int target) {
    for (int i = 0; i < numsSize; i++) {
        if (nums[i] >= target) {
            return i;
        }
    }
    return numsSize;
}
```

Accepted Runtime: 0 ms

```
nums =
[1,3,5,6]
target =
5
Output
2
```

203.Remove Linked List Elements



Accepted 66 / 66 testcases passed

RamithH submitted at Nov 30, 2025 23:40

Runtime 2 ms | Beats 20.21%

Memory 12.72 MB | Beats 15.21%

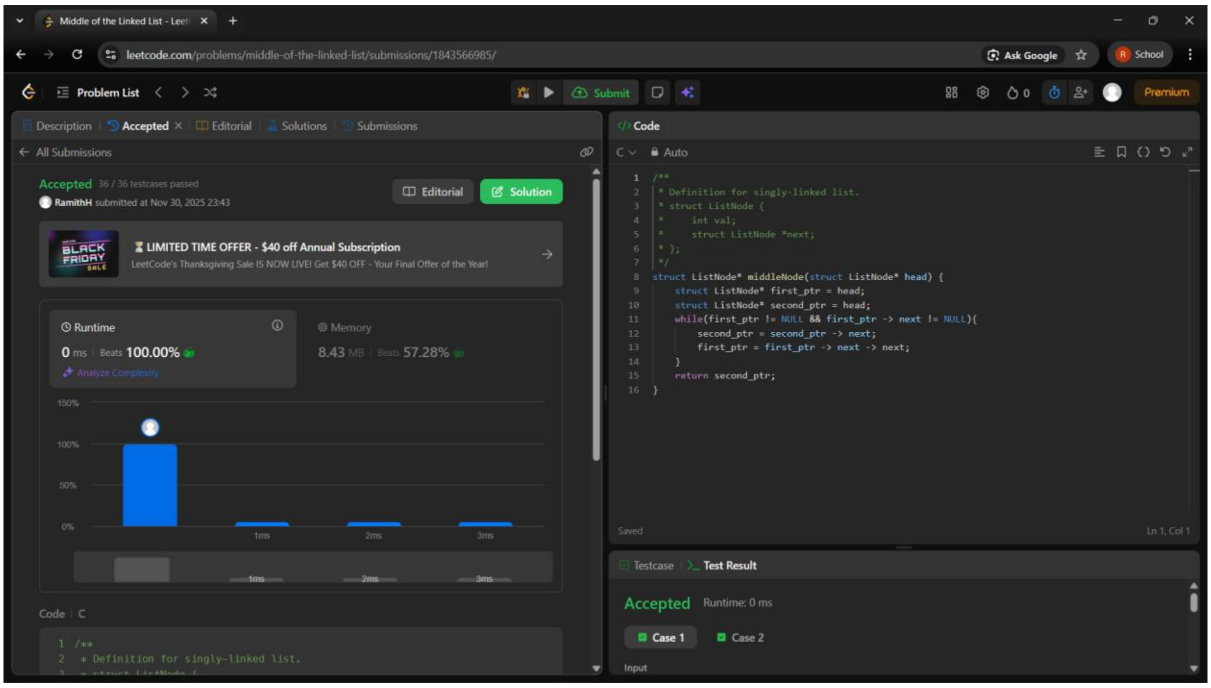
```c
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* removeElements(struct ListNode* head, int val) {
    struct ListNode dummy;
    dummy.next = head;
    struct ListNode* curr = &dummy;

    while(curr->next != NULL){
        if(curr->next->val == val){
            struct ListNode* temp = curr->next;
            curr->next = curr->next->next;
            free(temp);
        } else{
            curr = curr->next;
        }
    }

    return dummy.next;
}
```

Accepted Runtime: 0 ms

876.Middle of the Linked List



leetcode.com/problems/middle-of-the-linked-list/submissions/1843566985/

Accepted 36 / 36 testcases passed

RamithH submitted at Nov 30, 2025 23:43

LIMITED TIME OFFER - $40 off Annual Subscription

LeetCode's Thanksgiving Sale IS NOW LIVE! Get $40 OFF - Your Final Offer of the Year!

Runtime 0 ms | Beats 100.00%

Memory 8.43 MB | Beats 57.28%

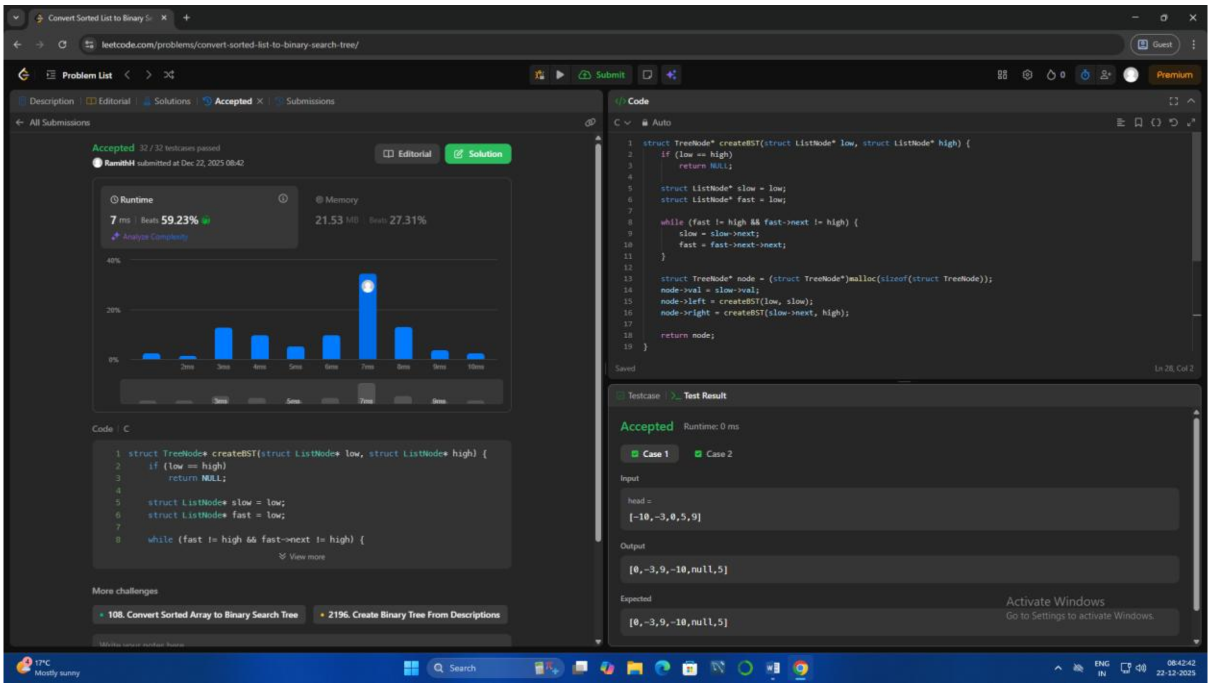
```c
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* middleNode(struct ListNode* head) {
    struct ListNode* first_ptr = head;
    struct ListNode* second_ptr = head;
    while(first_ptr != NULL && first_ptr -> next != NULL){
        second_ptr = second_ptr -> next;
        first_ptr = first_ptr -> next -> next;
    }
    return second_ptr;
}
```

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

109.Convert sorted list to Binary Search Tree



leetcode.com/problems/convert-sorted-list-to-binary-search-tree/

Accepted 32 / 32 testcases passed

RamithH submitted at Dec 22, 2025 08:42

Runtime 7 ms | Beats 59.23%

Memory 21.53 MB | Beats 27.31%

```c
struct TreeNode* createBST(struct ListNode* low, struct ListNode* high) {
    if (low == high)
        return NULL;

    struct ListNode* slow = low;
    struct ListNode* fast = low;

    while (fast != high && fast->next != high) {
        slow = slow->next;
        fast = fast->next->next;
    }

    struct TreeNode* node = (struct TreeNode*)malloc(sizeof(struct TreeNode));
    node->val = slow->val;
    node->left = createBST(low, slow);
    node->right = createBST(slow->next, high);

    return node;
}
```

More challenges

108. Convert Sorted Array to Binary Search Tree

2196. Create Binary Tree From Descriptions

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

head =

```
[-10,-3,0,5,9]
```

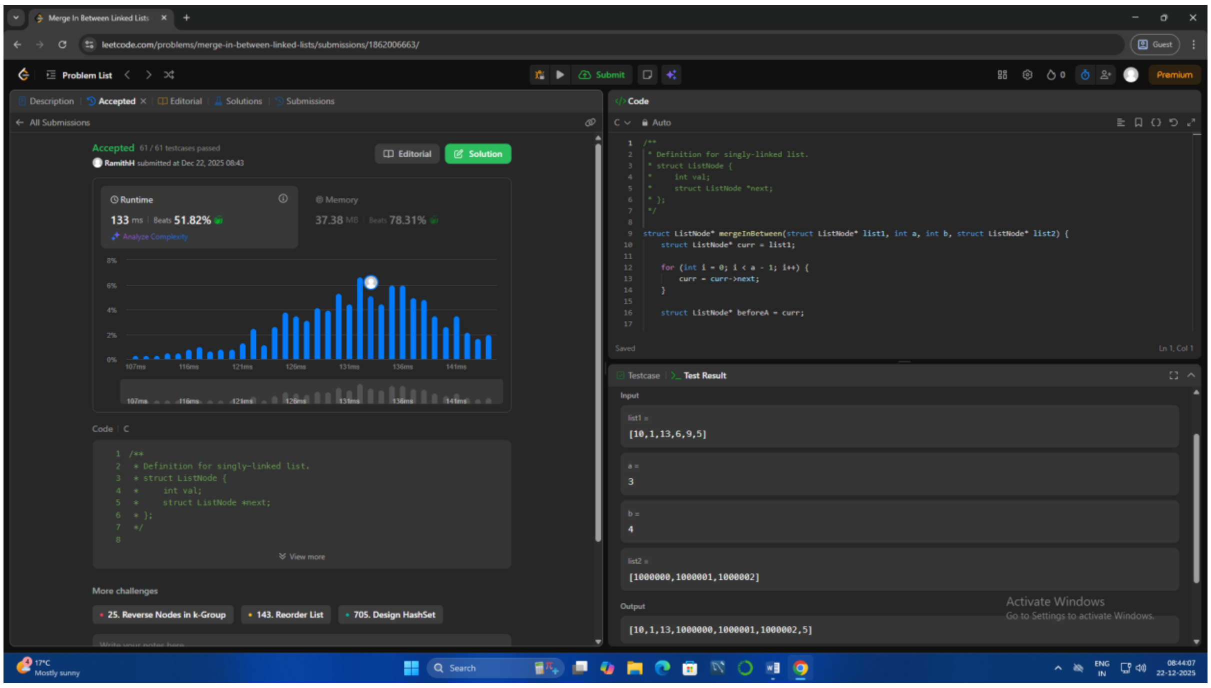
Output

```
[0,-3,9,-10,null,5]
```

Expected

```
[0,-3,9,-10,null,5]
```

1669.Merge in between linked lists



leetcode.com/problems/merge-in-between-linked-lists/submissions/1862006663/

Accepted 61 / 61 testcases passed

RamithH submitted at Dec 22, 2025 08:43

Runtime 133 ms | Beats 51.82%

Memory 37.38 MB | Beats 78.31%

Code | C

```c
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */

struct ListNode* mergeInBetween(struct ListNode* list1, int a, int b, struct ListNode* list2) {
    struct ListNode* curr = list1;

    for (int i = 0; i < a - 1; i++) {
        curr = curr->next;
    }

    struct ListNode* beforeA = curr;

```

More challenges: 25. Reverse Nodes in k-Group · 143. Reorder List · 705. Design HashSet

Input

```
list1 =
[10,1,13,6,9,5]
a =
3
b =
4
list2 =
[1000000,1000001,1000002]
```

Output

```
[10,1,13,1000000,1000001,1000002,5]
```