RAMAIAH INSTITUTE OF TECHNOLOGY

MSRIT NAGAR, BENGALURU, 560054

Dept. of ISE

A Survey Report On

[DATA STRUCTURES]

Submitted in partial fulfilment of the OTHER COMPONENT requirements as a part of the DATA STRUCTURES subject with code IS33 for the III Semester of degree of Bachelor of Engineering in Information Science and Engineering

Submitted by

SHRAVAN

1MS22IS128

Under the guidance of

Mr. Shivanand S

Assistant Professor

Dept. of ISE

Department of Information Science and Engineering

Ramaiah Institute of Technology

2023 – 2024

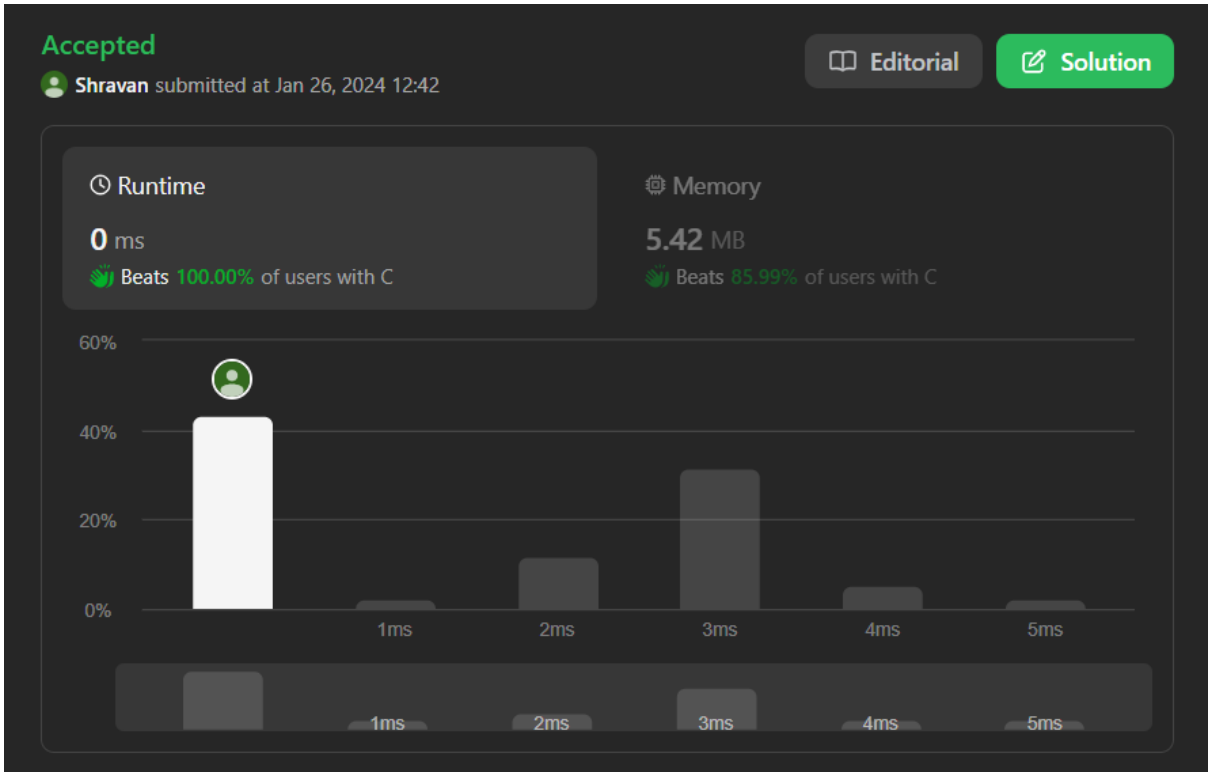
1. STACK

A. VALID PARENTHESES (#20 EASY)

```
bool isValid(char* bracket) {
    char st[10000];
    int top = -1;
    int i = 0;
    int length;
    length = strlen(bracket);
    if (length % 2 == 1)
        return 0;

    if (bracket[length - 1] == '(')
        return 0;
    else if (bracket[length - 1] == '{')
        return 0;
    else if (bracket[length - 1] == '[')
        return 0;

    while (bracket[i] != '\0') {
        char symbol = bracket[i++];
        if (symbol == '(') {
            st[++top] = symbol;

        } else if (symbol == '{') {
            st[++top] = symbol;

        } else if (symbol == '[') {
            st[++top] = symbol;

        } else if (symbol == ')') {
            if (top == -1)
                return 0;
            else {
                if (st[top] == '(')
                    st[top--];
                else
                    return 0;
            }
        } else if (symbol == '}') {
            if (top == -1)
                return 0;
            else {
                if (st[top] == '{')
                    st[top--];
                else
                    return 0;
            }
        } else if (symbol == ']') {
            if (i == 0)
```

```
                return 0;
            if (st[top] == '[')
                st[top--];
            else
                return 0;
        }
    }
    if (top == -1)
        return 1;
    else
        return 0;
}
```

OUTPUT:



Example 1:

Input: s = "()"

Output: true

Example 2:

Input: s = "()[]{}"

Output: true

Example 3:

Input: s = "(]"

Output: false

B. SCORE OF PARENTHESES (#856 MEDIUM)

```c
int scoreOfParentheses(char* s)
{
    int score = 0;
    int depth = 0;

    for (int i = 0; s[i] != '\0'; i++) {
        if (s[i] == '(')
        {
            depth++;
        }
        else
        {
            depth--;
            if (s[i - 1] == '(')
            {
                score += pow(2, depth);
            }
        }
    }
    return score;
}
```

OUTPUT:

Example 1:

Input: s = "()"

Output: 1

Example 2:

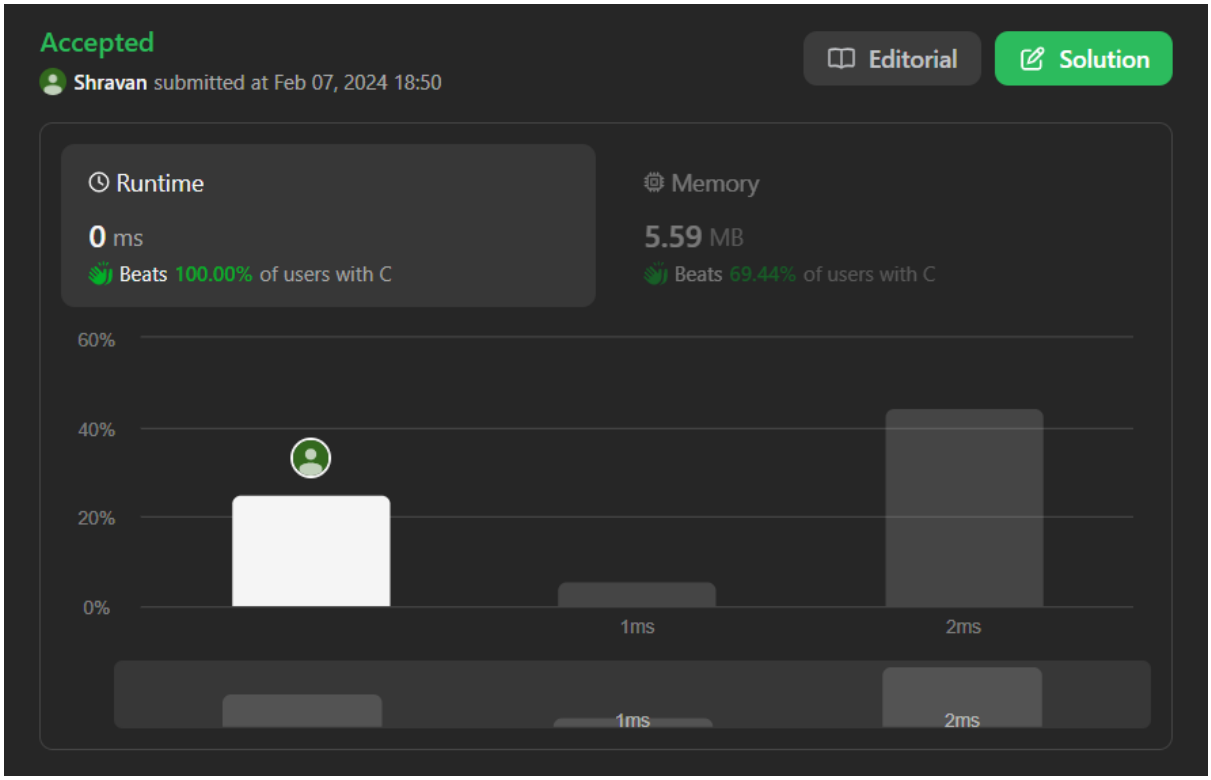
Input: s = "(())"

Output: 2

Example 3:

Input: s = "()()"

Output: 2



2. RECURSION

A. POWER OF TWO (#231 EASY)

```c
bool isPowerOfTwo(int n)
{
    if(n==1)
    return 1;
    if(n<1)
        return 0;
    return (n%2==0) && isPowerOfTwo(n/2);
}
```

OUTPUT:

Example 1:

Input: n = 1

Output: true

Explanation: $2^0 = 1$

Example 2:

Input: n = 16

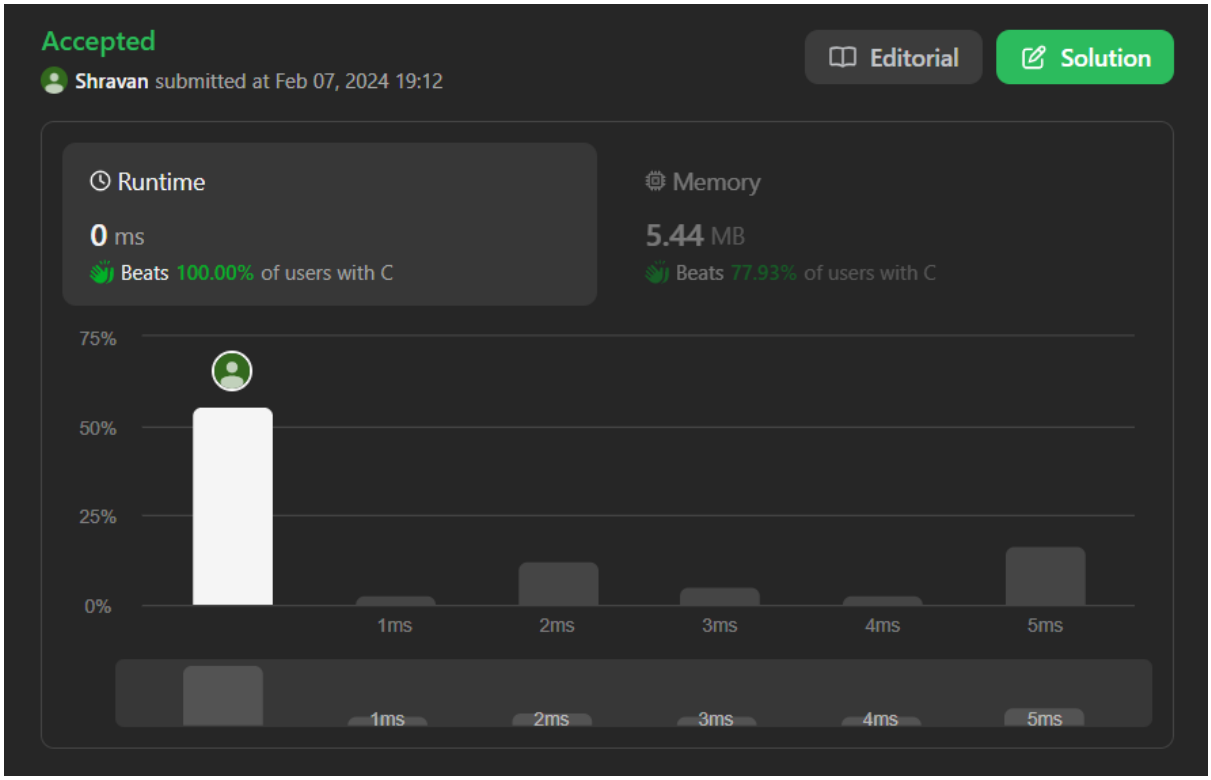
Output: true

Explanation: $2^4 = 16$

Example 3:

Input: n = 3

Output: false



B. POW(X,N) (#50 MEDIUM)

```c
double myPow(double x, int n) {
    long long int n1 = n;
    if (n1 == 0) {
        return 1;
    }
    if (n1 < 0) {
        n1 = -n1;
        x = 1 / x;
    }
    if (n1 % 2 == 0) {
        return myPow(x * x, n1 / 2);
    } else {
        return x * myPow(x * x, n1 / 2);
    }
}
```

OUTPUT:

Example 1:

Input: x = 2.00000, n = 10

Output: 1024.00000

Example 2:

Input: x = 2.10000, n = 3

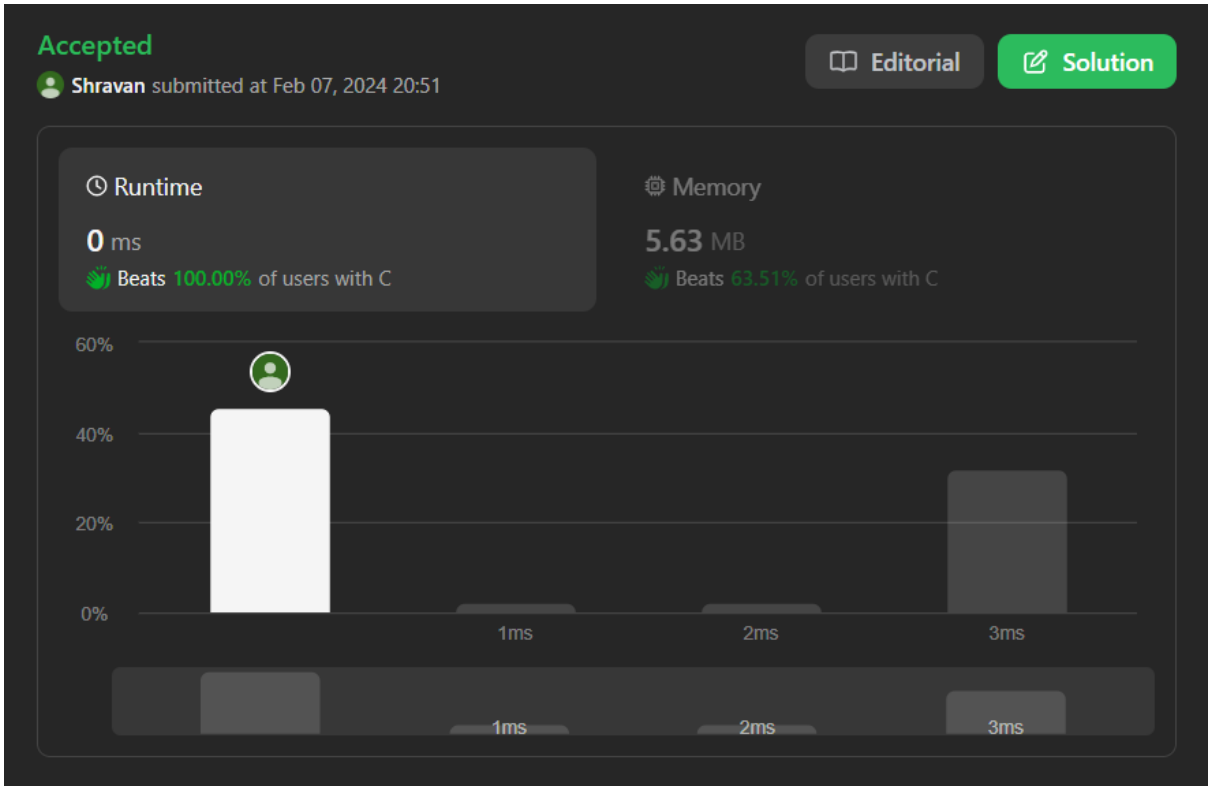
Output: 9.26100

Example 3:

Input: x = 2.00000, n = -2

Output: 0.25000

Explanation: $2^{-2} = 1/2^{2} = 1/4 = 0.25$

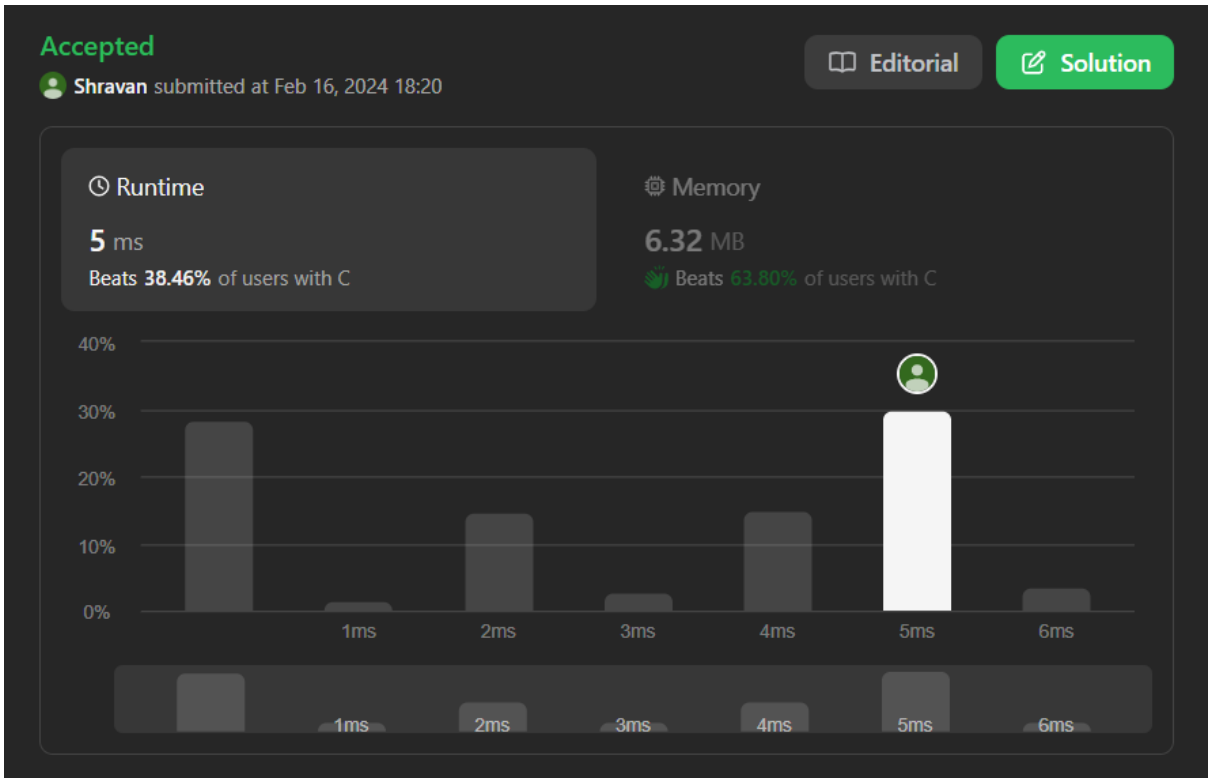


3. LINKED LIST

A. REVERSE LINKED LIST (#206 EASY)

```c
struct ListNode* reverseList(struct ListNode* head)
{
    int q[5000];
    if (head == NULL)
        return head;
    struct ListNode *temp=head;
    int i=0;
    while(temp!=NULL)
    {
        q[i]=temp->val;
        temp=temp->next;
        i++;
    }
    struct ListNode *cur=head;
    while((i-1)>=0)
    {
        cur->val=q[i-1];
        cur=cur->next;
        i--;
    }
    return head;
}
```

OUTPUT:



Example 1:

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

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

Example 2:

Input: head = [1,2]

Output: [2,1]

Example 3:

Input: head = []

Output: []

B. DELETE A NODE IN ALINKED LIST (#237 MEDIUM)

```
void deleteNode(struct ListNode* node) {

    node->val=node->next->val;
    node->next=node->next->next;

}
```

OUTPUT:

Input: head = [4,5,1,9], node = 5

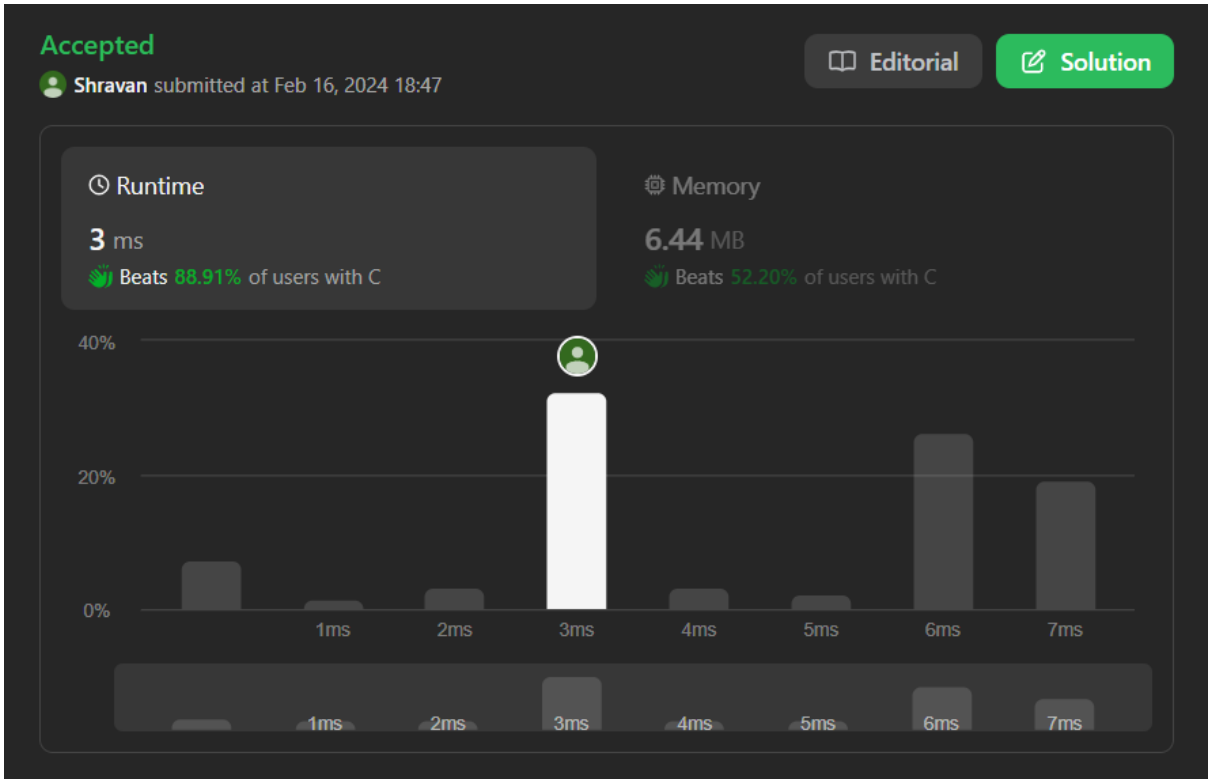
Output: [4,1,9]

Explanation: You are given the second node with value 5, the linked list should become 4 -> 1 –> 9 after calling your function.

Input: head = [4,5,1,9], node = 1

Output: [4,5,9]

Explanation: You are given the third node with value 1, the linked list should become 4 -> 5 -> 9 after calling your function.



4. QUEUE

A. FIRST UNIQUE CHARACTER IN A STRING (#387 EASY)

```c
int firstUniqChar(char* s) {
    int count[26] = {0};
    for (int i = 0; s[i] != '\0'; i++) {
        count[s[i] - 'a'] += 1;
    }
    for (int i = 0; s[i] != '\0'; i++) {
        if (count[s[i] - 'a'] == 1) {
            return i;
        }
    }
    return -1;
}
```

OUTPUT;

Example 1:

Input: s = "leetcode"

Output: 0

Example 2:

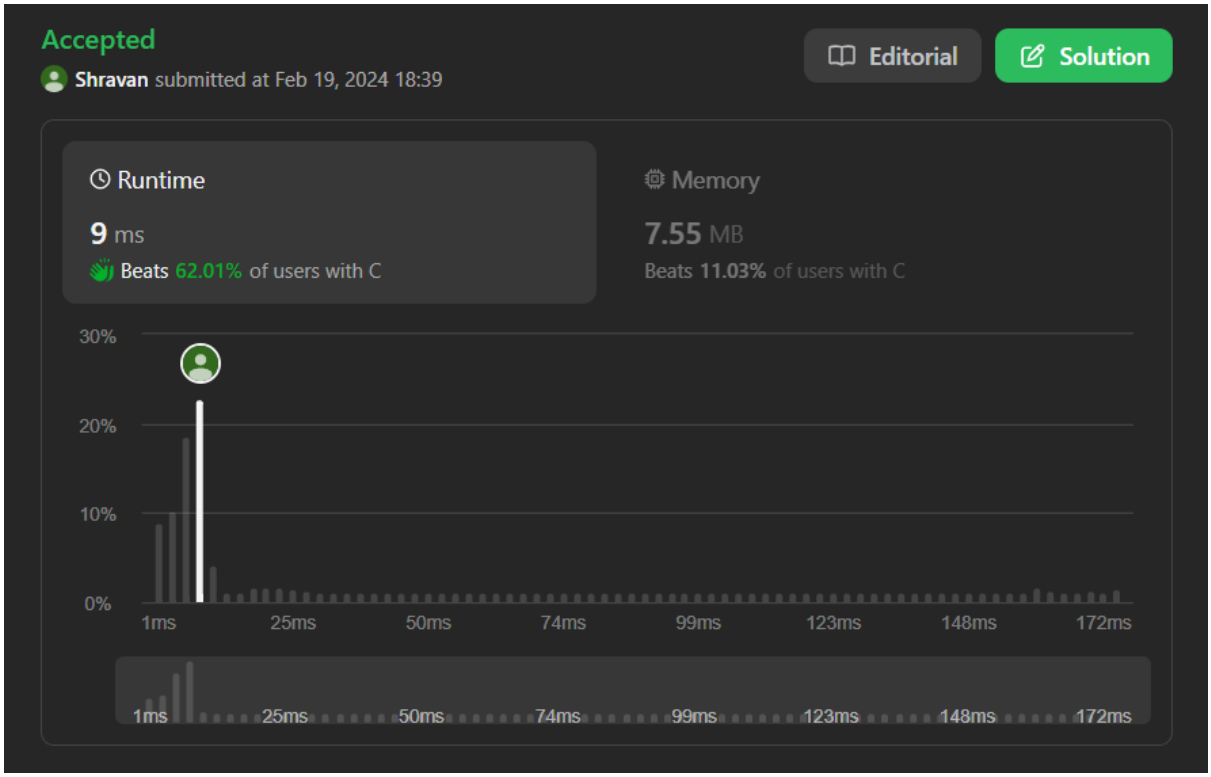
Input: s = "loveleetcode"

Output: 2

Example 3:

Input: s = "aabb"

Output: -1



B. FIND THE WINNER OF THE CIRCULAR GAME (#1823 MEDIUM)

```
int findTheWinner(int n, int k)
{
    int winner=0;
    for (int i=1;i<= n;i++)
        winner=(winner+k)%i;
    return winner+1;
}
```

OUTPUT:

Example 1:

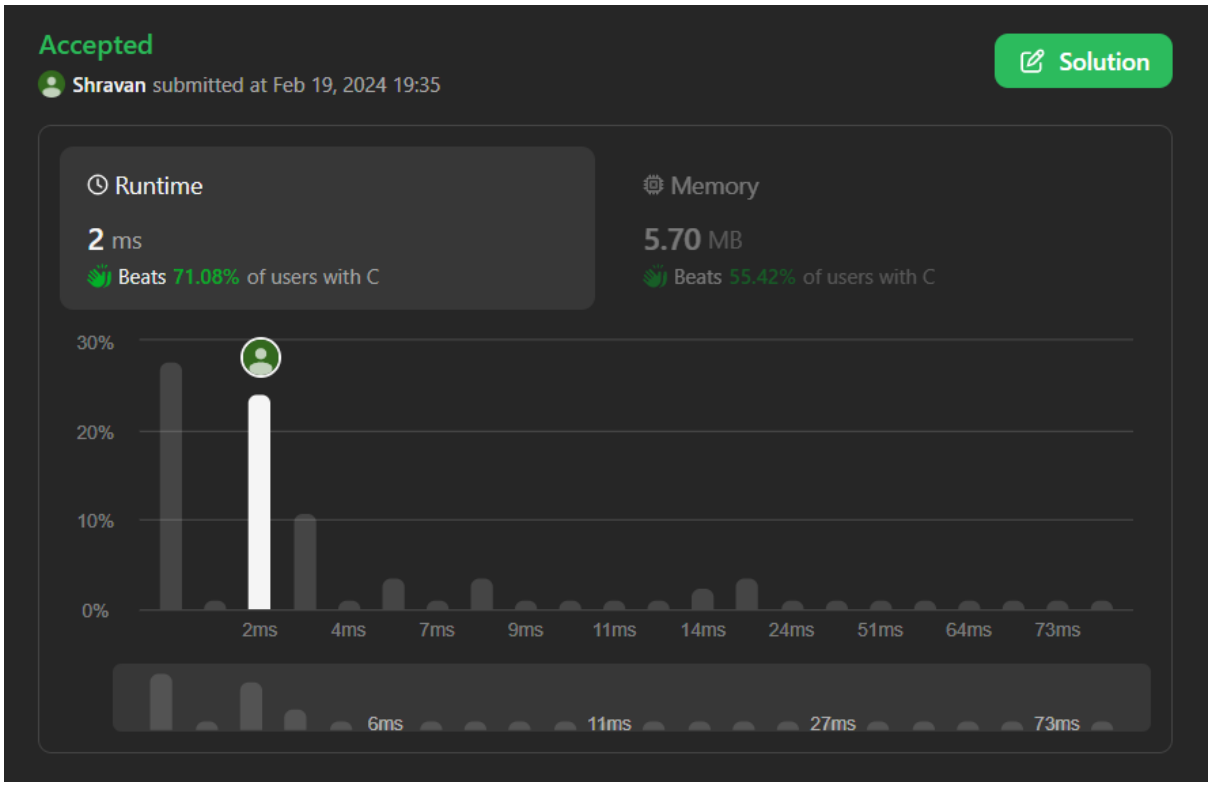
Input: n = 5, k = 2

Output: 3

Example 2:

Input: n = 6, k = 5

Output: 1



5. BINARY SEARCH TREE

A. CONVERT SORTED ARRAY TO BST (#108 EASY)

```c
struct TreeNode* sortedArrayToBST(int* nums, int numsSize)
{
    if (numsSize == 0) return NULL;

    int mid = numsSize / 2;

    struct TreeNode* root = (struct TreeNode*)malloc(sizeof(struct TreeNode));
    root->val = nums[mid];
    root->left = sortedArrayToBST(nums, mid);
    root->right = sortedArrayToBST(nums + mid + 1, numsSize - mid - 1);
    return root;
}
```

OUTPUT:

Example 1:

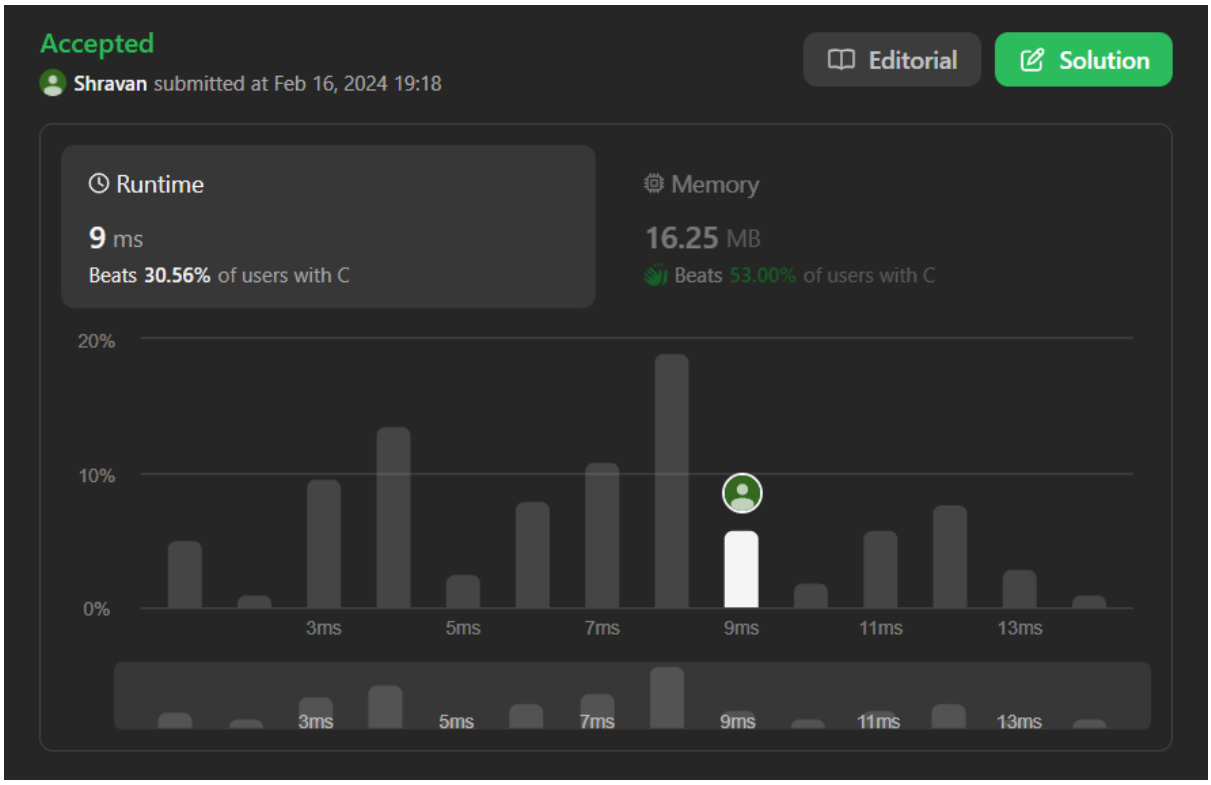
Input: nums = [-10,-3,0,5,9]

Output: [0,-3,9,-10,null,5]

Example 2:

Input: nums = [1,3]

Output: [3,1]



B. VALIDATE BINARY SEARCH TREE (#98 MEDIUM)

```c
int st[10000];
void inorder(struct TreeNode *root, int *top);
bool isValidBST(struct TreeNode* root)
{
    if(root==NULL)
        return true;

    int top=-1;
    inorder(root,&top);

    for(int i=1;i<=top;i++)
    {
        if(st[i]<=st[i-1])
            return false;
    }
    return true;
```

```c
}
void inorder(struct TreeNode *root,int *top)
{
    if(root==NULL)
        return;
    inorder(root->left,top);
    st[++(*top)]=root->val;
    inorder(root->right,top);
}
```

OUTPUT:

Example 1:

Input: root = [2,1,3]

Output: true

Example 2:

Input: root = [5,1,4,null,null,3,6]

Output: false