

Winter Domain Camp Day-3

Name-Abhiraj Patel

UID-22BCS11329

Date-23-12-24

Q1 .Fibonnacci Series Using Recursion

The screenshot shows a submission for a Fibonacci series problem. The submission is marked as "Accepted" and was submitted by Abhiraj Patel on Dec 23, 2024, at 15:20. The runtime is 15 ms, which beats 10.09% of other submissions. The memory usage is 7.60 MB, which beats 50.58% of other submissions. The code is written in C++ and uses a recursive helper function to calculate the Fibonacci sequence.

```
1 class Solution {
2 public:
3     int helper(int n)
4     {
5         if(n==0) return 0;
6         if(n==1) return 1;
7         return helper(n-1)+helper(n-2);
8     }
9     int fib(int n) {
10         return helper(n);
11     }
12 };
```

2.To Find Reverse Of String Using Recursion

The screenshot shows a submission for reversing a string using recursion. The submission is marked as "Accepted" and was submitted by Abhiraj Patel on Dec 23, 2024, at 15:24. The runtime is 0 ms, which beats 100.00% of other submissions. The memory usage is 27.20 MB, which beats 84.46% of other submissions. The code is written in C++ and uses a recursive helper function to reverse the string by swapping the first and last characters and then recursively reversing the remaining part of the string.

```
1 class Solution {
2 public:
3     void helper(vector<char>&s,int l,int r)
4     {
5         if(l>r) return;
6         swap(s[l],s[r]);
7         helper(s,l+1,r-1);
8     }
9     void reverseString(vector<char>& s) {
10         helper(s,0,s.size()-1);
11     }
12 };
```

3. Merge Two Sorted Lists

Accepted

Abhiraj Patel submitted at Dec 23, 2024 15:25

Editorial Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

19.36 MB | Beats 94.29%

100%

```
9  * );
10 */
11 class Solution {
12 public:
13     ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
14         if(list1==NULL) return list2;
15         if(list2==NULL) return list1;
16         if(list1->val<list2->val)
17         {
18             list1->next=mergeTwoLists(list1->next,list2);
19             return list1;
20         }
21         else {
22             list2->next=mergeTwoLists(list1,list2->next);
23             return list2;}
24     }
25 };
```

4. Remove Linked List Elements

Accepted

Abhiraj Patel submitted at Dec 23, 2024 15:28

Editorial Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

20.47 MB | Beats 6.52%

100%

```
1 class Solution {
2 public:
3     ListNode* removeElements(ListNode* head, int val) {
4         if (head == nullptr) return nullptr;
5         head->next = removeElements(head->next, val);
6         return head->val == val ? head->next : head;
7     }
8 };
```

5. Reverse Linked List

Accepted

Abhiraj Patel submitted at Dec 23, 2024 15:29

Editorial Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

13.73 MB | Beats 7.82%

```
8     ListNode(int x, ListNode *next) : val(x), next
9     * );
10 */
11 class Solution {
12 public:
13     ListNode* reverseList(ListNode* head) {
14         if(!head||!head->next) return head;
15         ListNode*revl=reverseList(head->next);
16         ListNode*front=head->next;
17         front->next=head;
18         head->next=NULL;
19         return revl;
20     }
21 };
```