Day 30

Code-

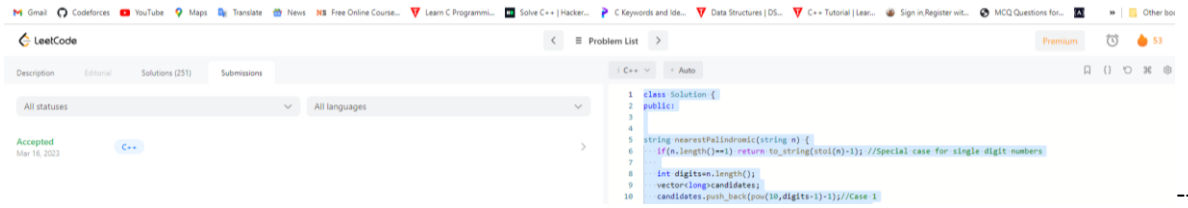
```
class Solution {
public:


string nearestPalindromic(string n) {
   if(n.length()==1) return to_string(stoi(n)-1); //Special case for single digit
numbers

   int digits=n.length();
   vector<long>candidates;
   candidates.push_back(pow(10,digits-1)-1);//Case 1
   candidates.push_back(pow(10,digits)+1);//Case 2

   int mid=(digits+1)/2;
   long prefix=stol(n.substr(0,mid));
   //Case 3
   vector<long>v={prefix,prefix+1,prefix-1};
   for(long i:v)
   {
       string postfix=to_string(i);
       if(digits%2!=0) postfix.pop_back();//// If the total length is odd number,
pop the middle number in postfix
       reverse(postfix.begin(),postfix.end());
       string c=to_string(i)+postfix;
       candidates.push_back(stol(c));
   }
   long mindiff=LONG_MAX;long result;long num=stol(n);
   for(int i=0;i<5;i++)
   {
       if(candidates[i]!=num&&abs(candidates[i]-num)<mindiff)//Candidate must not
be the same number and abs diff is minm
       {
           mindiff=abs(candidates[i]-num);
           result=candidates[i];
       }
       else if(abs(candidates[i]-num)==mindiff) result=min(result,candidates[i]);
   }
    return to_string(result);
}
};
```

Time Complexity -O(N)
Space complexity- O(n)


Gmail
Codeforces
YouTube
Maps
Translate
News
Free Online Course...
Learn C Programmi...
Solve C++ | Hacker...
C Keywords and Ide...
Data Structures | DS...
C++ Tutorial | Lear...
Sign in,Register wit...
MCQ Questions for...
Other bo
LeetCode
Problem List
Premium
53
Description
Editorial
Solutions (251)
Submissions
All statuses
All languages
Accepted
Mar 16, 2023
C++
C++
Auto
1 class Solution {
2 public:
3
4
5 string nearestPalindromic(string n) {
6 if(n.length()==1) return to_string(stoi(n)-1); //Special case for single digit numbers
7
8 int digits=n.length();
9 vector<long> candidates;
10 candidates.push_back(pow(10,digits-1)-1);//Case 1