DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Experiment-1.4

Student Name: Kanishk Soni **UID: 20BCS9398**
Branch: BE-CSE **Section/Group:20BCS_DM-708B**
Semester: 6th **Subject Name: Competitive Coding-II**
Subject Code: 20CSP-351

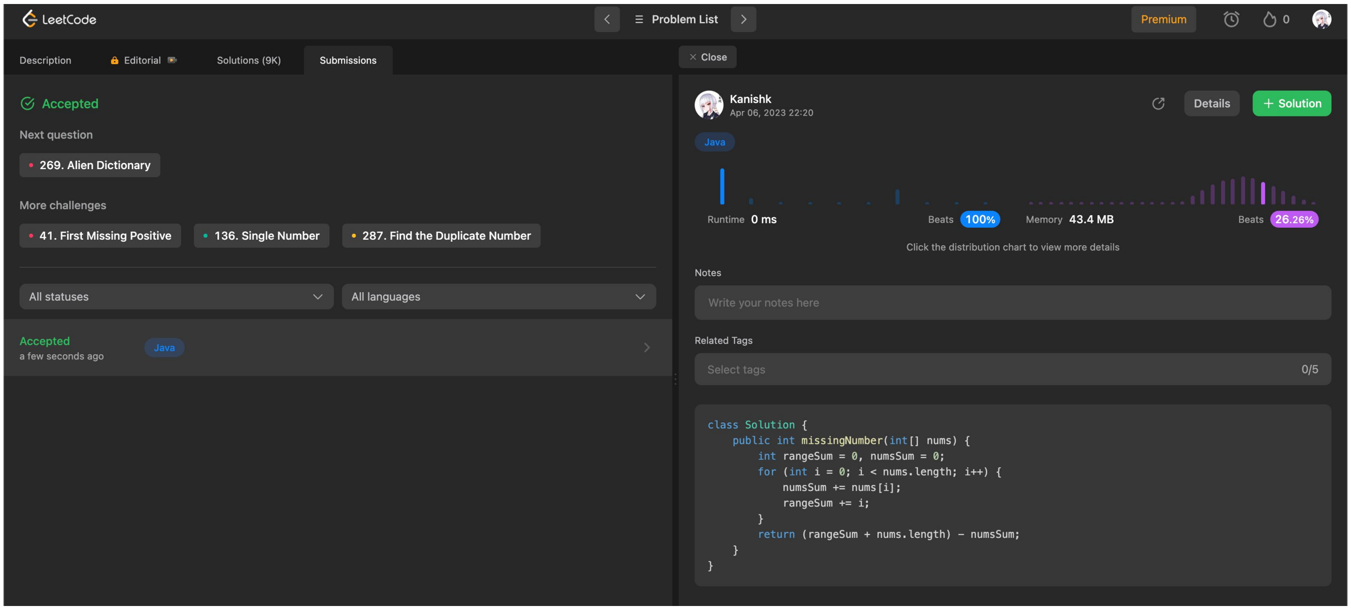
AIM: To demonstrate the concept of Hashing.

Problem1: Missing Number
https://leetcode.com/problems/missing-number/

Program Code:

```
class Solution {
    public int missingNumber(int[] nums) {
        int rangeSum = 0, numsSum = 0;
        for (int i = 0; i < nums.length; i++) {
            numsSum += nums[i];
            rangeSum += i;
        }
        return (rangeSum + nums.length) – numsSum;
    }
}
```

Output:



Accepted

Accepted
a few seconds ago

Kanishk
Apr 06, 2023 22:20

Runtime 0 ms Beats 100%

Memory 43.4 MB Beats 26.26%

Click the distribution chart to view more details

```
class Solution {
    public int missingNumber(int[] nums) {
        int rangeSum = 0, numsSum = 0;
        for (int i = 0; i < nums.length; i++) {
            numsSum += nums[i];
            rangeSum += i;
        }
        return (rangeSum + nums.length) - numsSum;
    }
}
```

Probem2: Longest duplicate substring
https://leetcode.com/problems/longest-duplicate-substring/

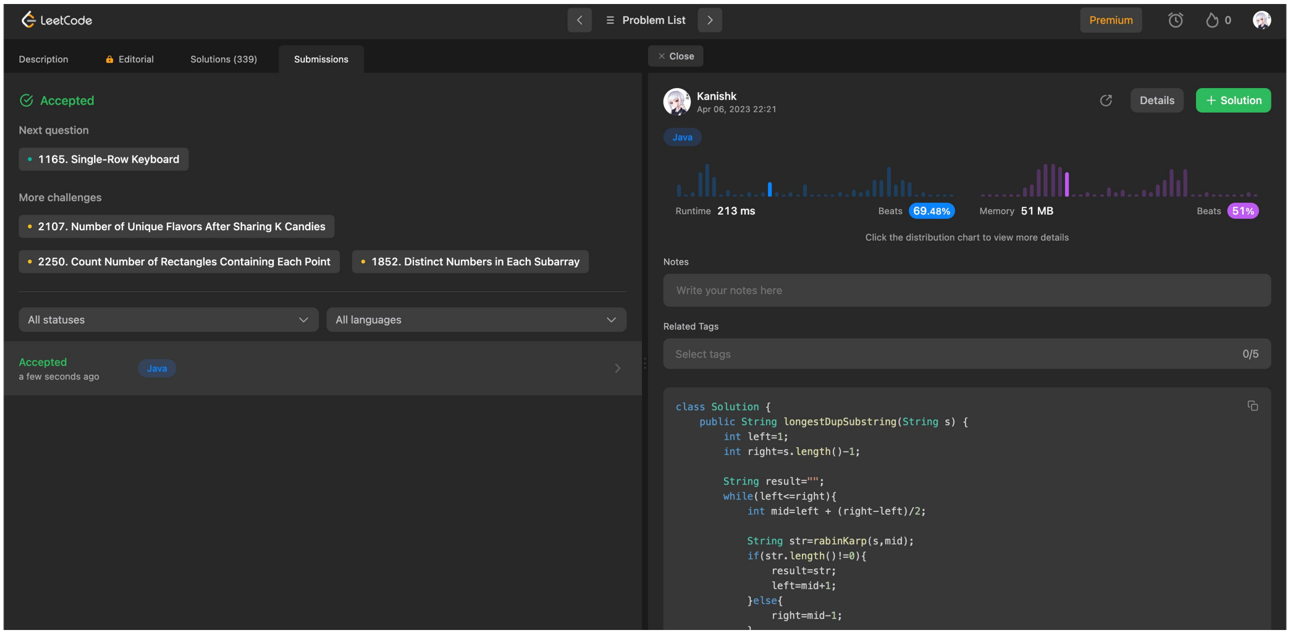
Program Code:

```
class Solution {
    public String longestDupSubstring(String s) {
        int left=1;
        int right=s.length()-1;
        String result="";
        while(left<=right){
            int mid=left + (right-left)/2;
            String str=rabinKarp(s,mid);
            if(str.length()!=0){
                result=str;
                left=mid+1;
            }else{
                right=mid–1;
            }
        }
        return result;
    }
    private String rabinKarp(String s,int len){
        Set<Long> set=new HashSet<>();
        long h=hash(s.substring(0,len));
        set.add(h);
        long pow=1;
        for(int l=1;l<len;l++) pow*=31;
        for(int i=1;i<=s.length()–len;i++){
            h=nextHash(pow,h,s.charAt(i–1),s.charAt(i+len–1));
            if(set.contains(h)){
                return s.substring(i,i+len);
            }
            set.add(h);
        }
        return "";
    }

    private long nextHash(long pow,long hash,char left,char right){
        return (hash – left*pow)*31 + right;
        // abcd   bcdf
```

```
  }
   private long hash(String s) {
                    long hash = 0;
        long mul=1;
                    for (int i = s.length()–1; i >=0; i––) {
                           hash += s.charAt(i)*mul;
           mul*=31;
                    }
                    return hash;
         }
}
```

Output:



Accepted

Next question

- 1165. Single-Row Keyboard

More challenges

- 2107. Number of Unique Flavors After Sharing K Candies
- 2250. Count Number of Rectangles Containing Each Point
- 1852. Distinct Numbers in Each Subarray

Accepted
a few seconds ago

Kanishk
Apr 06, 2023 22:21

Runtime 213 ms Beats 69.48%
Memory 51 MB Beats 51%

Click the distribution chart to view more details

Notes

Write your notes here

Related Tags

Select tags 0/5

```
class Solution {
    public String longestDupSubstring(String s) {
        int left=1;
        int right=s.length()-1;

        String result="";
        while(left<=right){
            int mid=left + (right-left)/2;

            String str=rabinKarp(s,mid);
            if(str.length()!=0){
                result=str;
                left=mid+1;
            }else{
                right=mid-1;
            }
```