

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Experiment-1.4

Student Name: Kanishk Soni
Branch: BE-CSE
Semester: 6th
Subject Code: 20CSP-351

UID: 20BCS9398
Section/Group: 20BCS_DM-708B
Subject Name: Competitive Coding-II

AIM: To demonstrate the concept of Hashing.

Problem1: Missing Number

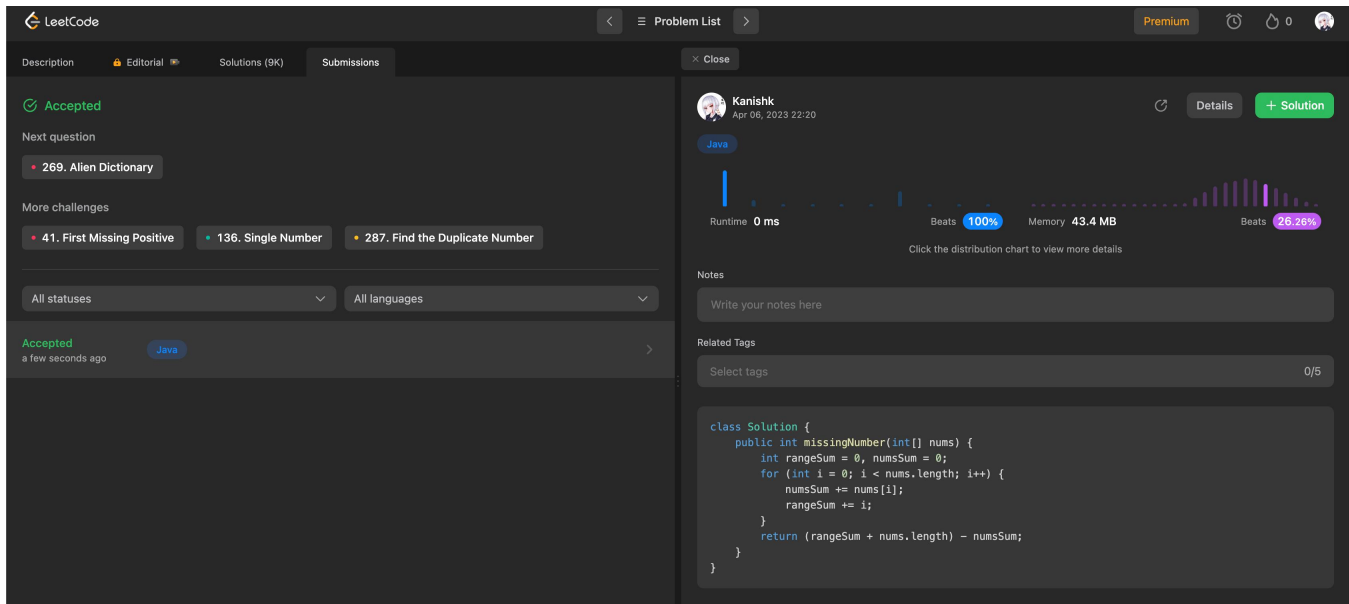
<https://leetcode.com/problems/missing-number/>

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;  
    }  
}
```

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Output:



The screenshot shows the LeetCode interface for the problem 'Find the Duplicate Number' (Problem 287). The submission is 'Accepted' and was made by user 'Kanishk' on April 06, 2023. The code is in Java. Performance metrics show a runtime of 0 ms, 100% beats, and memory usage of 43.4 MB. The code implements a solution using a range sum and a nums sum to find the duplicate number.

```
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;  
    }  
}
```

Problem2: 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;  
            }  
        }  
        return result;  
    }  
}
```

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
        }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
```

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
}  
  
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:

The screenshot shows a LeetCode submission interface. On the left, the 'Submissions' tab is active, showing a green 'Accepted' status and a 'Next question' button. Below this, there are links to other problems: '1165. Single-Row Keyboard', '2107. Number of Unique Flavors After Sharing K Candies', '2250. Count Number of Rectangles Containing Each Point', and '1852. Distinct Numbers in Each Subarray'. A dropdown menu for 'All statuses' is set to 'Accepted', and a dropdown for 'All languages' is set to 'Java'. The submission was made 'a few seconds ago'.

On the right, the 'Details' tab is active, showing the user 'Kanishk' (Apr 06, 2023 22:21) and the 'Java' language. The submission details include a runtime of 213 ms (Beats 69.48%) and memory usage of 51 MB (Beats 51%). A distribution chart is visible, with a note to 'Click the distribution chart to view more details'. Below the chart, there is a 'Notes' section with a text input field and a 'Related Tags' section with a 'Select tags' dropdown (0/5).

The solution code is displayed in a dark-themed editor:

```
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;  
    }  
}
```