Experiment1.4

Student Nam Manan Bhardwaj **UID** 21BCS7131
Branch: Be-Cse **Section/Group:** CC 645-A
Semester: 6th **Date of Performan** 07-02-2024
Subject Name: Ap Lab-2 **Subject Code:** 21CSP-351

1. Aim:

a) To Find the single missing number in an array containing all consecutive integers from 0 to n excluding one number.

b) To Find the longest substring within a string that appears at least twice, allowing overlaps.

2. Objective:

a) Given an array nums containing n distinct numbers in the range [0, n], return the only number in the range that is missing from the array.

b) Given a string s, consider all duplicated substrings: (contiguous) substrings of s that occur 2 or more times. The occurrences may overlap.

Return **any** duplicated substring that has the longest possible length. If s does not have a duplicated substring, the answer is "".

3. Code (A):

```
#include <vector>
#include <numeric>

class Solution { public:
   int missingNumber(vector<int>& nums) {
// Step 1: Find the length of the array.
      int n = nums.size();

      // Step 2: Calculate the actual sum using mathematical formula.
```

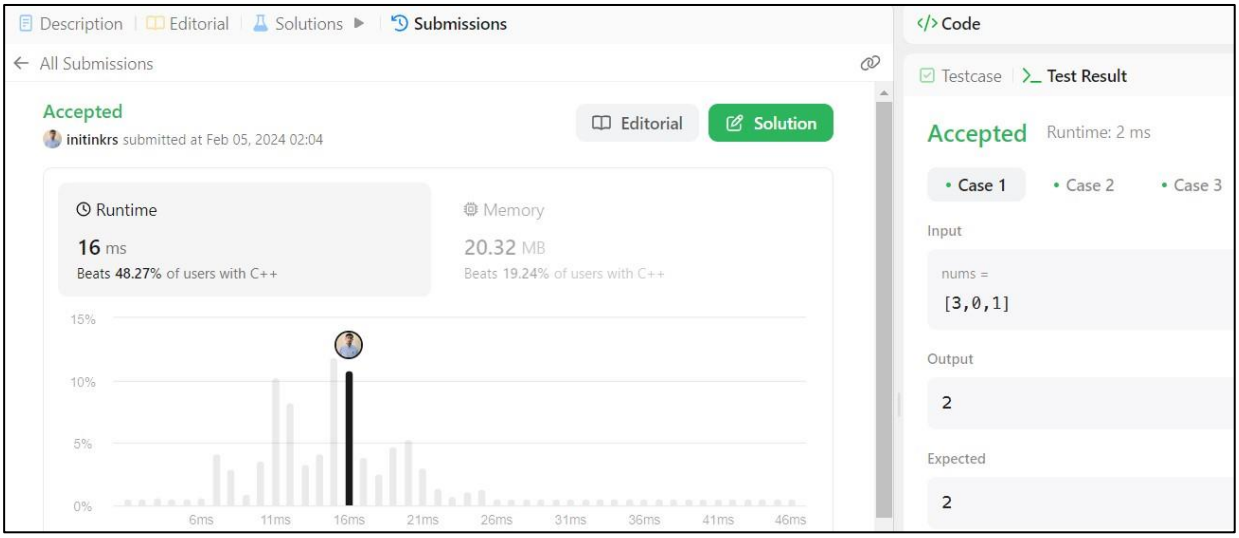
```
        int actualSum = n * (n + 1) / 2;

        // Step 3: Calculate the given sum using accumulate.
    int sum = accumulate(nums.begin(), nums.end(), 0);

        // Step 4: Return the difference between the actual and given sums.
    return actualSum - sum;
      }
    };
```

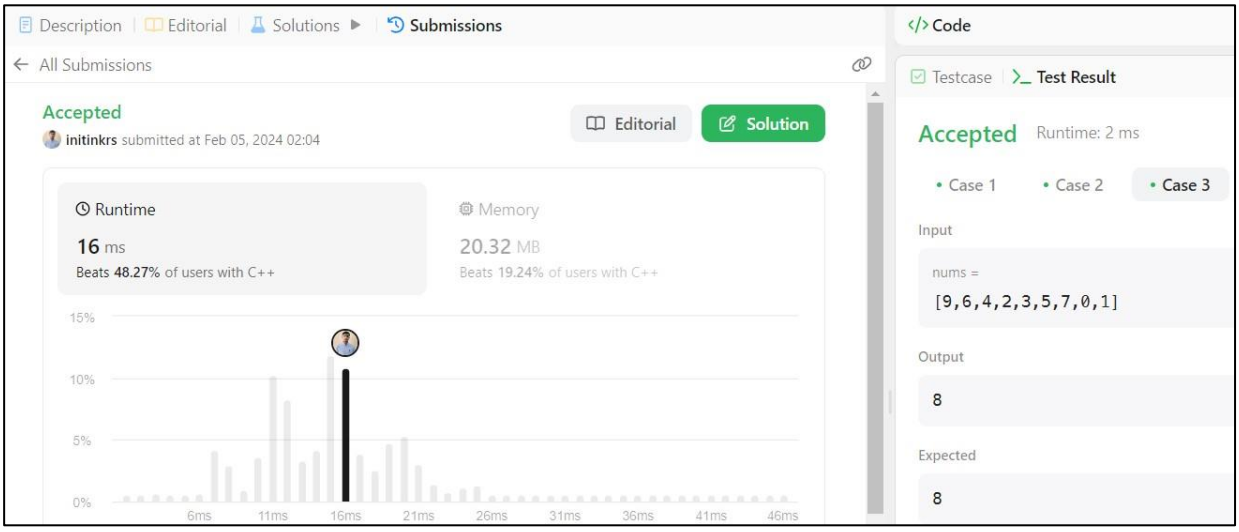
Code (B):

```
class Solution { public:          string
longestDupSubstring(string S)
   {
      std::string_view longest;
      std::unordered_set<std::string_view> set;
      size_t beg = 1;      size_t
end = S.size() - 1;
      while (beg <= end)
      {
         auto len = beg + (end - beg) / 2;
bool found = false;
         for (size_t i = 0; i != S.size() - len + 1; ++i)
         {
            const auto [it, inserted] = set.emplace(S.data() + i, len);
if (!inserted)
            {
               found = true;
longest          =          *it;
break;
            }
         }
                  if
(found)              beg =
```

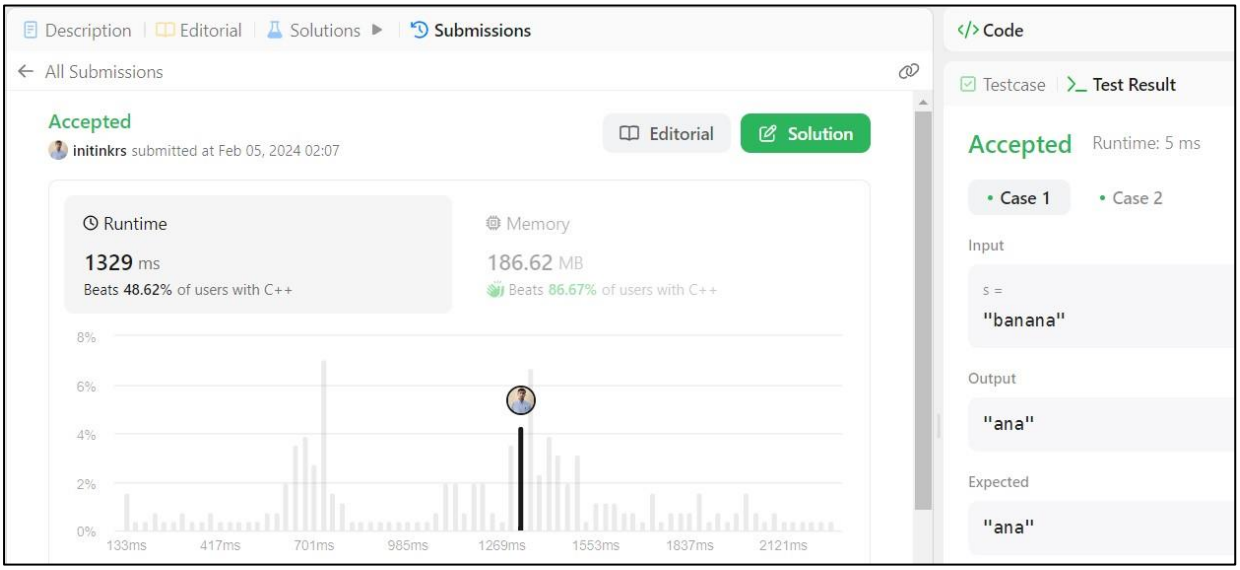
```
len + 1;            else
end = len - 1;

            set.clear();
        }
        return {longest.begin(), longest.end()};
    }
};
```

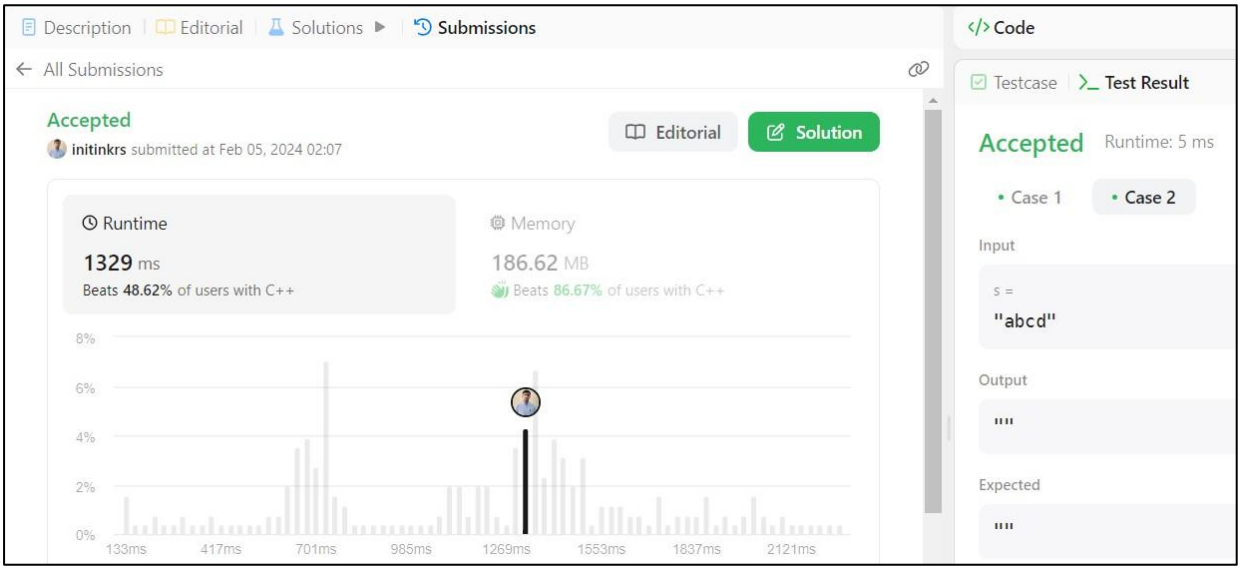
4. Output (A):





Output (B):


Description
Editorial
Solutions
Submissions
All Submissions
Accepted
initinkrs submitted at Feb 05, 2024 02:07
Editorial
Solution
Runtime
1329 ms
Beats 48.62% of users with C++
Memory
186.62 MB
Beats 86.67% of users with C++
8%
6%
4%
2%
0%
133ms
417ms
701ms
985ms
1269ms
1553ms
1837ms
2121ms
Code
Testcase
Test Result
Accepted
Runtime: 5 ms
Case 1
Case 2
Input
s =
"banana"
Output
"ana"
Expected
"ana"


Description
Editorial
Solutions
Submissions
All Submissions
Accepted
initinkrs submitted at Feb 05, 2024 02:07
Editorial
Solution
Runtime
1329 ms
Beats 48.62% of users with C++
Memory
186.62 MB
Beats 86.67% of users with C++
8%
6%
4%
2%
0%
133ms
417ms
701ms
985ms
1269ms
1553ms
1837ms
2121ms
Code
Testcase
Test Result
Accepted
Runtime: 5 ms
Case 1
Case 2
Input
s =
"abcd"
Output
""
Expected
""