Experiment: 2

Student Name: Kapil Gangwar Uid: 22BCS50113
Branch: BE-CSE Section: 22BCS_IOT_627(A)
Semester: 6th Date of Performance: Jan 31, 2025
Subject: Advanced Programming Lab-2 Subject Code: 22CSP-351

1. Aim:

Problem 2.1: Two Sum

Problem Statement: Given an array of integers nums and an integer target, return the indices of the two numbers such that they add up to target. Each input has exactly one solution, and you cannot use the same element twice.

Problem 2.2: Jump Game II

Problem Statement: You are given a 0-indexed array nums of length n. You are initially positioned at nums[0]. Each element nums[i] represents the maximum length of a forward jump from index i. Return the minimum number of jumps to reach nums[n - 1].

Problem 2.3: Simplify Path

Problem Statement: Given a string path, which is an absolute path to a file or directory in a Unix-style file system, convert it to the simplified canonical path.

2. Objective:

- To Implement the concepts of Array, Stacks, and Queues.

3. Implementation / Code:

2.1:

```
class Solution {
public:
   vector<int> twoSum(vector<int>& nums, int target) {
      unordered_map<int, int> num_map;
      for (int i = 0; i < nums.size(); ++i) {
         int complement = target - nums[i];
         if (num_map.find(complement) != num_map.end()) {
            return {num_map[complement], i};
         }
         num_map[nums[i]] = i;
      }
      return {};
   }
   };
```

2.2:

```
class Solution {
public:
   int jump(vector<int>& nums) {
      int jumps = 0, currentEnd = 0, farthest = 0;
      for (int i = 0; i < nums.size() - 1; ++i) {
         farthest = max(farthest, i + nums[i]);
         if (i == currentEnd) {
            jumps++;
            currentEnd = farthest;
          }
        }
        return jumps;
      }
      };
```

2.3:

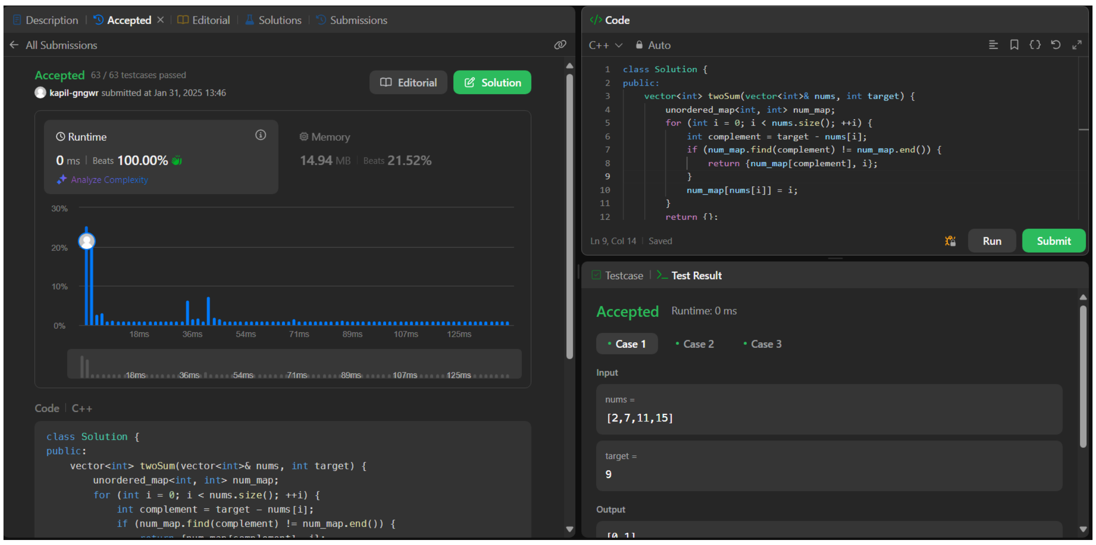
```
class Solution {
public:
    string simplifyPath(string path) {
        vector<string> stack;
        stringstream ss(path);
        string token;

        while (getline(ss, token, '/')) {
            if (token == "" || token == ".") continue;
            if (token == "..") {
                if (!stack.empty()) stack.pop_back();
            } else {
                stack.push_back(token);
            }
        }

        string result = "/";
        for (int i = 0; i < stack.size(); ++i) {
            result += stack[i];
            if (i != stack.size() - 1) result += "/";
        }

        return result;
    }
    };
```

4. Output:



Accepted 63 / 63 testcases passed
kapil-gngwr submitted at Jan 31, 2025 13:46

Runtime: 0 ms | Beats 100.00%
Memory: 14.94 MB | Beats 21.52%

Code | C++

```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        unordered_map<int, int> num_map;
        for (int i = 0; i < nums.size(); ++i) {
            int complement = target - nums[i];
            if (num_map.find(complement) != num_map.end()) {
                return {num_map[complement], i};
            }
            num_map[nums[i]] = i;
        }
        return {};
```

Ln 9, Col 14 | Saved

Accepted Runtime: 0 ms

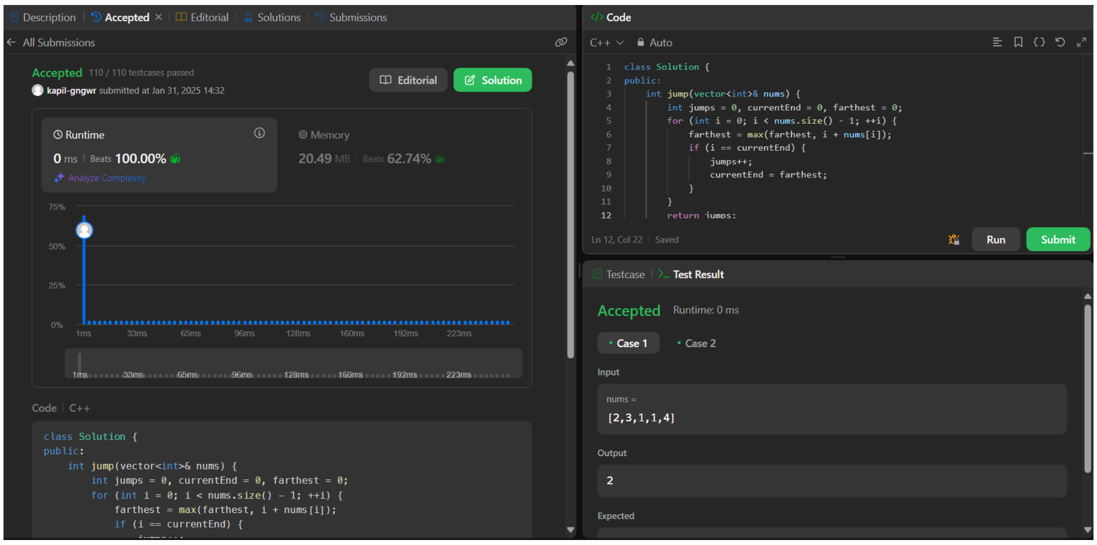
Input

nums =
[2,7,11,15]

target =
9

Output

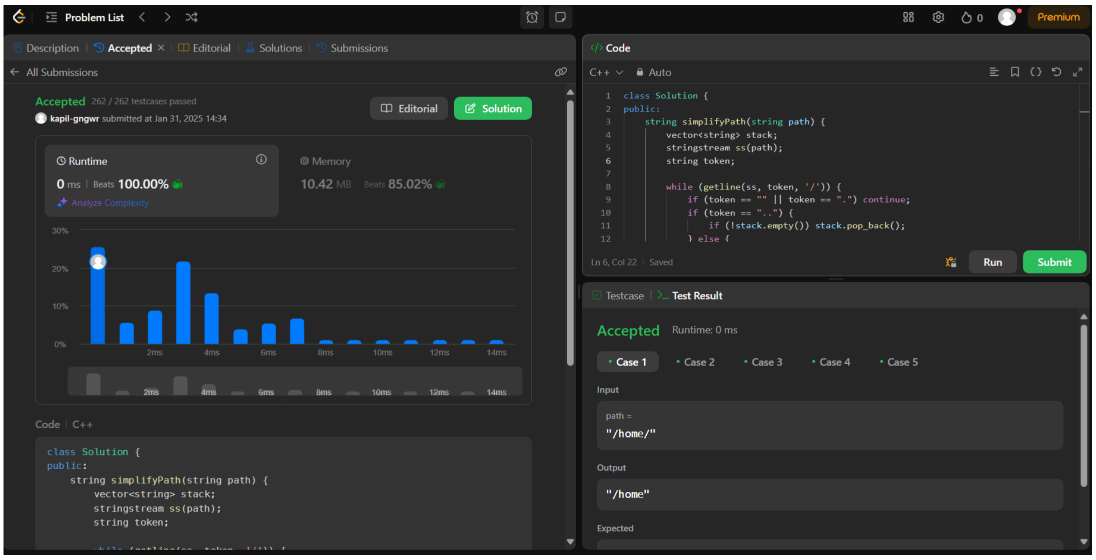
[0,1]


Accepted 110 / 110 testcases passed
kapil-gngwr submitted at Jan 31, 2025 14:32
Runtime 0 ms Beats 100.00%
Memory 20.49 MB Beats 62.74%

```
class Solution {
public:
    int jump(vector<int>& nums) {
        int jumps = 0, currentEnd = 0, farthest = 0;
        for (int i = 0; i < nums.size() - 1; ++i) {
            farthest = max(farthest, i + nums[i]);
            if (i == currentEnd) {
                jumps++;
                currentEnd = farthest;
            }
        }
        return jumps;
```

Ln 12, Col 22 | Saved

Accepted Runtime: 0 ms
Case 1 Case 2
Input
nums =
[2,3,1,1,4]
Output
2
Expected


Accepted 262 / 262 testcases passed
kapil-gngwr submitted at Jan 31, 2025 14:34
Runtime 0 ms Beats 100.00%
Memory 10.42 MB Beats 85.02%

```
class Solution {
public:
    string simplifyPath(string path) {
        vector<string> stack;
        stringstream ss(path);
        string token;

        while (getline(ss, token, '/')) {
            if (token == "" || token == ".") continue;
            if (token == "..") {
                if (!stack.empty()) stack.pop_back();
            } else {
```

Ln 6, Col 22 | Saved

Accepted Runtime: 0 ms
Case 1 Case 2 Case 3 Case 4 Case 5
Input
path =
"/home/"
Output
"/home"
Expected

5. Learning Outcome:

- Merge Two Sorted Lists: Learned how to recursively merge two sorted singly linked lists.
- Two Sum: Learned how to use a hash map to find two numbers that add up to a target value.
- Jump Game II: Learned how to use a greedy algorithm to find the minimum number of jumps to reach the end of an array.
- Simplify Path: Learned how to use a stack to simplify a Unix-style file path.