

DSA Assignment - 20 June

Name: Aditya Tiwari

Roll No: 23/11/EC/040

LeetCode ID: Aditya1305T

GeeksForGeeks ID: aditya1305t

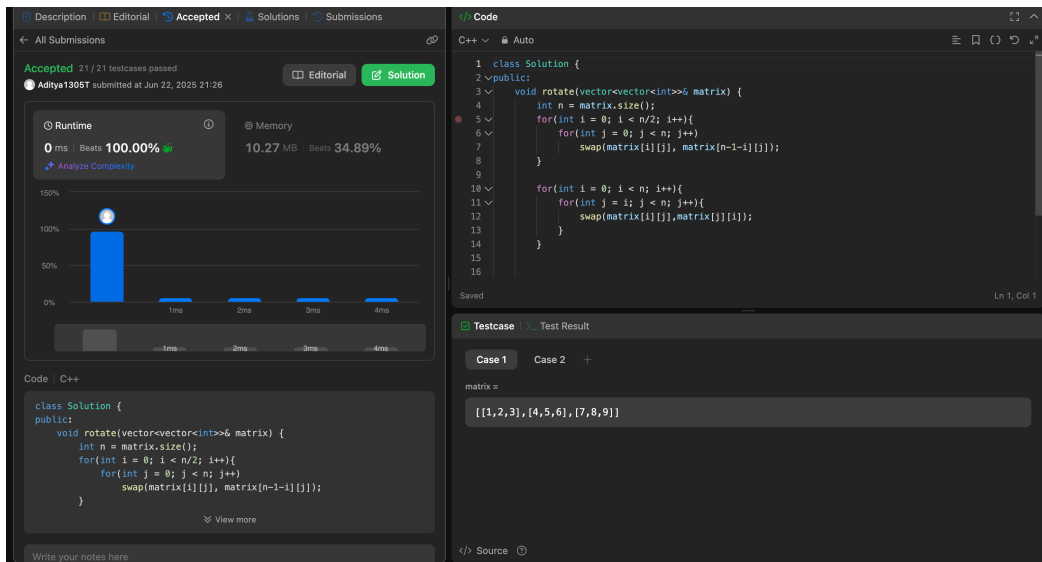
Github Repo Link:

https://github.com/Aditya1305T/SOE_Training_25

Question 1: Rotate Image

Platform: LeetCode

Link: <https://leetcode.com/problems/rotate-image/>



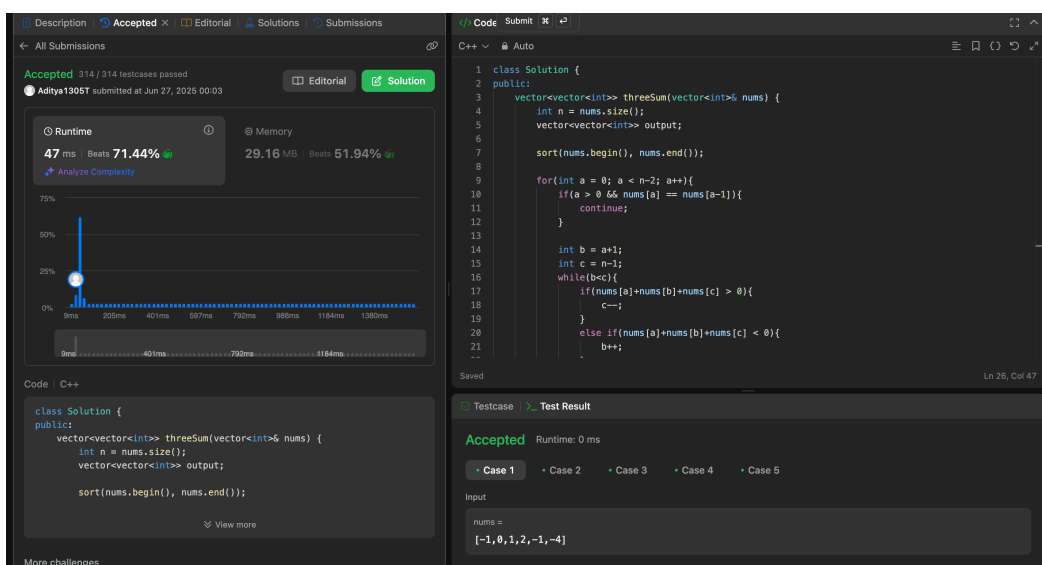
The screenshot shows the LeetCode interface for the 'Rotate Image' problem. The left sidebar displays the 'Accepted' status with 21/21 testcases passed, submitted at Jun 22, 2025 21:26. The runtime is 0 ms, beating 100.00%, and memory is 10.27 MB, beating 34.89%. A bar chart shows the runtime distribution. The main area contains the C++ code for the 'rotate' function, which uses nested loops to swap elements in a matrix. The right sidebar shows the 'Testcase' tab with Case 1 selected, displaying the input matrix: `[[1,2,3],[4,5,6],[7,8,9]]`.

```
class Solution {
public:
    void rotate(vector<vector<int>>& matrix) {
        int n = matrix.size();
        for(int i = 0; i < n/2; i++){
            for(int j = 0; j < n; j++){
                swap(matrix[i][j], matrix[n-1-i][j]);
            }
        }
        for(int i = 0; i < n; i++){
            for(int j = 1; j < n; j++){
                swap(matrix[i][j], matrix[i][j-1]);
            }
        }
    }
};
```

Question 2: 3 Sum

Platform: LeetCode

Link: <https://leetcode.com/problems/3sum/description/>



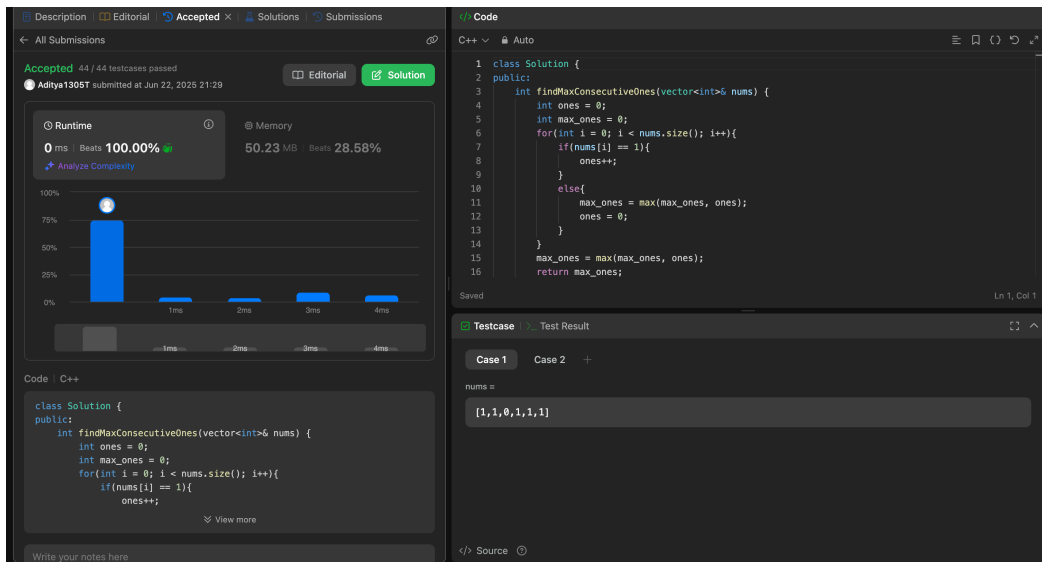
The screenshot shows the LeetCode interface for the '3 Sum' problem. The left sidebar displays the 'Accepted' status with 314/314 testcases passed, submitted at Jun 27, 2025 00:03. The runtime is 47 ms, beating 71.44%, and memory is 29.16 MB, beating 51.94%. A bar chart shows the runtime distribution. The main area contains the C++ code for the 'threeSum' function, which sorts the input array and uses a two-pointer technique to find triplets that sum to zero. The right sidebar shows the 'Testcase' tab with Case 1 selected, displaying the input array: `nums = [-1,0,1,2,-1,-4]`.

```
class Solution {
public:
    vector<vector<int>> threeSum(vector<int>& nums) {
        int n = nums.size();
        vector<vector<int>> output;
        sort(nums.begin(), nums.end());
        for(int a = 0; a < n-2; a++){
            if(a > 0 && nums[a] == nums[a-1]){
                continue;
            }
            int b = a+1;
            int c = n-1;
            while(b < c){
                if(nums[a]+nums[b]+nums[c] > 0){
                    c--;
                }
                else if(nums[a]+nums[b]+nums[c] < 0){
                    b++;
                }
                else{
                    output.push_back({nums[a], nums[b], nums[c]});
                    b++;
                    c--;
                }
            }
        }
        return output;
    }
};
```

Question 3: Maximum Consecutive Ones

Platform: LeetCode

Link: <https://leetcode.com/problems/max-consecutive-ones/>



Question 4: Design Twitter

Platform: LeetCode

Link: <https://leetcode.com/problems/design-twitter/description/>

