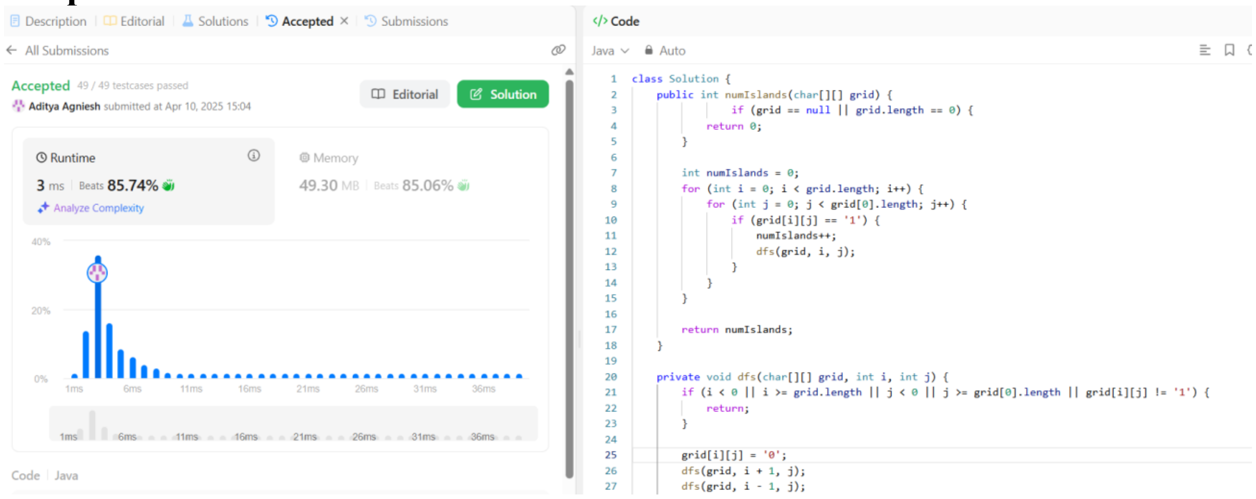
Experiment 9

Student Name: Aditya Agniesh **UID:22BCS14644**
Branch: CSE **Section/Group:614_B**
Semester: 6th **Date of Performance:10/04/25**
Subject Name: AP lAB **Subject Code: 22CSP-351**

Ques 1:Number of Islands

Code:
```
class Solution {
    public int numIslands(char[][] grid) {
                if (grid == null || grid.length == 0) {
            return 0;
        }

        int numIslands = 0;
        for (int i = 0; i < grid.length; i++) {
            for (int j = 0; j < grid[0].length; j++) {
                if (grid[i][j] == '1') {
                    numIslands++;
                    dfs(grid, i, j);
                }
            }
        }

        return numIslands;
    }

    private void dfs(char[][] grid, int i, int j) {
        if (i < 0 || i >= grid.length || j < 0 || j >= grid[0].length || grid[i][j] != '1') {
            return;
        }

        grid[i][j] = '0';
        dfs(grid, i + 1, j);
        dfs(grid, i - 1, j);
        dfs(grid, i, j + 1);
        dfs(grid, i, j - 1);
    }

    }
```

Output


Description | Editorial | Solutions | Accepted × | Submissions
All Submissions
Editorial
Solution
Code
Java
Auto

Accepted 49 / 49 testcases passed
Aditya Agniesh submitted at Apr 10, 2025 15:04

Runtime
3 ms | Beats 85.74%
Analyze Complexity

Memory
49.30 MB | Beats 85.06%

Code | Java

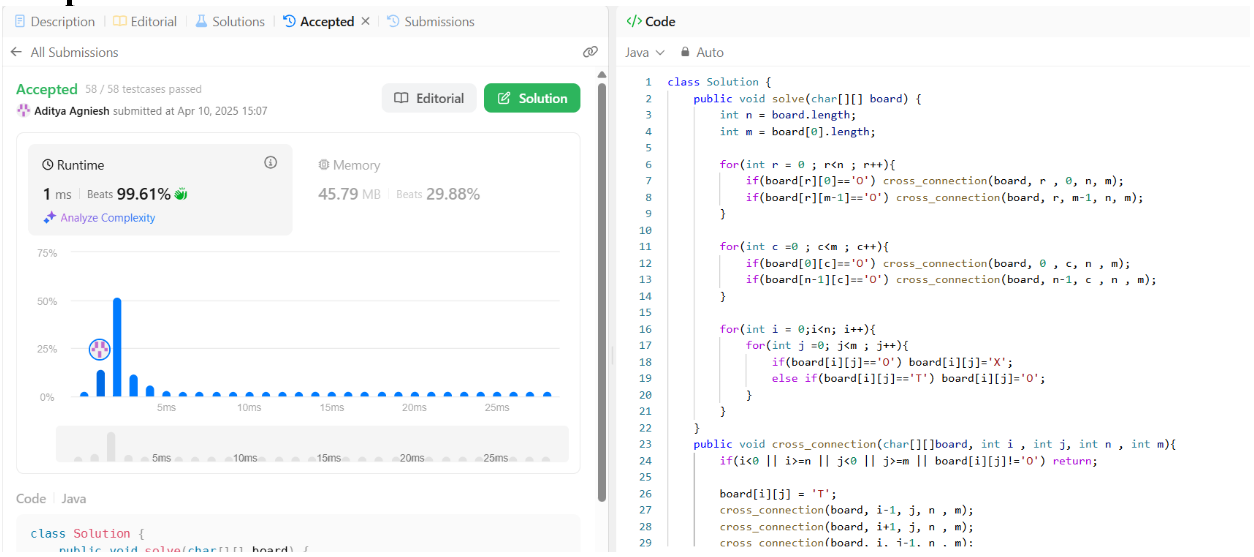
```
class Solution {
    public int numIslands(char[][] grid) {
            if (grid == null || grid.length == 0) {
        return 0;
    }

    int numIslands = 0;
    for (int i = 0; i < grid.length; i++) {
        for (int j = 0; j < grid[0].length; j++) {
            if (grid[i][j] == '1') {
                numIslands++;
                dfs(grid, i, j);
            }
        }
    }

    return numIslands;
    }

    private void dfs(char[][] grid, int i, int j) {
        if (i < 0 || i >= grid.length || j < 0 || j >= grid[0].length || grid[i][j] != '1') {
            return;
        }

        grid[i][j] = '0';
        dfs(grid, i + 1, j);
        dfs(grid, i - 1, j);
```

Ques 2:Surrounded Regions

Code

```
class Solution {
    public void solve(char[][] board) {
        int n = board.length;
        int m = board[0].length;

        for(int r = 0 ; r<n ; r++){
            if(board[r][0]=='O') cross_connection(board, r , 0, n, m);
            if(board[r][m-1]=='O') cross_connection(board, r, m-1, n, m);
        }

        for(int c =0 ; c<m ; c++){
            if(board[0][c]=='O') cross_connection(board, 0 , c, n , m);
          if(board[n-1][c]=='O') cross_connection(board, n-1, c , n , m);
        }

        for(int i = 0;i<n; i++){
            for(int j =0; j<m ; j++){
                if(board[i][j]=='O') board[i][j]='X';
                else if(board[i][j]=='T') board[i][j]='O';
            }
        }
    }
    public void cross_connection(char[][]board, int i , int j, int n , int m){
        if(i<0 || i>=n || j<0 || j>=m || board[i][j]!='O') return;

        board[i][j] = 'T';
        cross_connection(board, i-1, j, n , m);
        cross_connection(board, i+1, j, n , m);
        cross_connection(board, i, j-1, n , m);
        cross_connection(board, i, j+1, n , m);
    }
}
```

Output:



Accepted 58 / 58 testcases passed
Aditya Agniesh submitted at Apr 10, 2025 15:07

Runtime
1 ms | Beats 99.61%

Memory
45.79 MB | Beats 29.88%

Code | Java

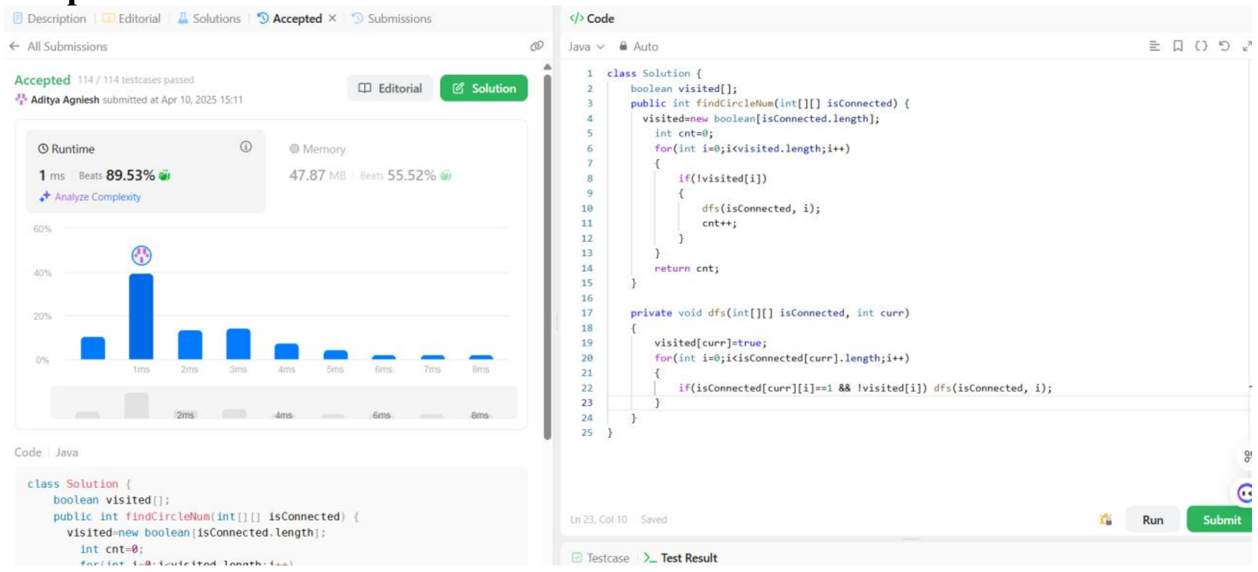
```
class Solution {
    public void solve(char[][] board) {
        int n = board.length;
        int m = board[0].length;

        for(int r = 0 ; r<n ; r++){
            if(board[r][0]=='O') cross_connection(board, r , 0, n, m);
            if(board[r][m-1]=='O') cross_connection(board, r, m-1, n, m);
        }

        for(int c =0 ; c<m ; c++){
            if(board[0][c]=='O') cross_connection(board, 0 , c, n , m);
            if(board[n-1][c]=='O') cross_connection(board, n-1, c , n , m);
        }

        for(int i = 0;i<n; i++){
            for(int j =0; j<m ; j++){
                if(board[i][j]=='O') board[i][j]='X';
                else if(board[i][j]=='T') board[i][j]='O';
            }
        }
    }
    public void cross_connection(char[][]board, int i , int j, int n , int m){
        if(i<0 || i>=n || j<0 || j>=m || board[i][j]!='O') return;

        board[i][j] = 'T';
        cross_connection(board, i-1, j, n , m);
        cross_connection(board, i+1, j, n , m);
        cross_connection(board, i, j-1, n , m);
```

Ques 3: Friend Circle

Code:

```
class Solution {
    boolean visited[];
    public int findCircleNum(int[][] isConnected) {
      visited=new boolean[isConnected.length];
        int cnt=0;
        for(int i=0;i<visited.length;i++)
        {
            if(!visited[i])
            {
                dfs(isConnected, i);
                cnt++;
            }
        }
        return cnt;
    }

    private void dfs(int[][] isConnected, int curr)
    {
        visited[curr]=true;
        for(int i=0;i<isConnected[curr].length;i++)
        {
            if(isConnected[curr][i]==1 && !visited[i]) dfs(isConnected, i);
        }
    }
}
```

Output:



Accepted 114 / 114 testcases passed
Aditya Agniesh submitted at Apr 10, 2025 15:11

Runtime
1 ms | Beats 89.53%

Memory
47.87 MB | Beats 55.52%

```
class Solution {
    boolean visited[];
    public int findCircleNum(int[][] isConnected) {
      visited=new boolean[isConnected.length];
        int cnt=0;
        for(int i=0;i<visited.length;i++)
        {
            if(!visited[i])
            {
                dfs(isConnected, i);
                cnt++;
            }
        }
        return cnt;
    }

    private void dfs(int[][] isConnected, int curr)
    {
        visited[curr]=true;
        for(int i=0;i<isConnected[curr].length;i++)
        {
            if(isConnected[curr][i]==1 && !visited[i]) dfs(isConnected, i);
        }
    }
}
```

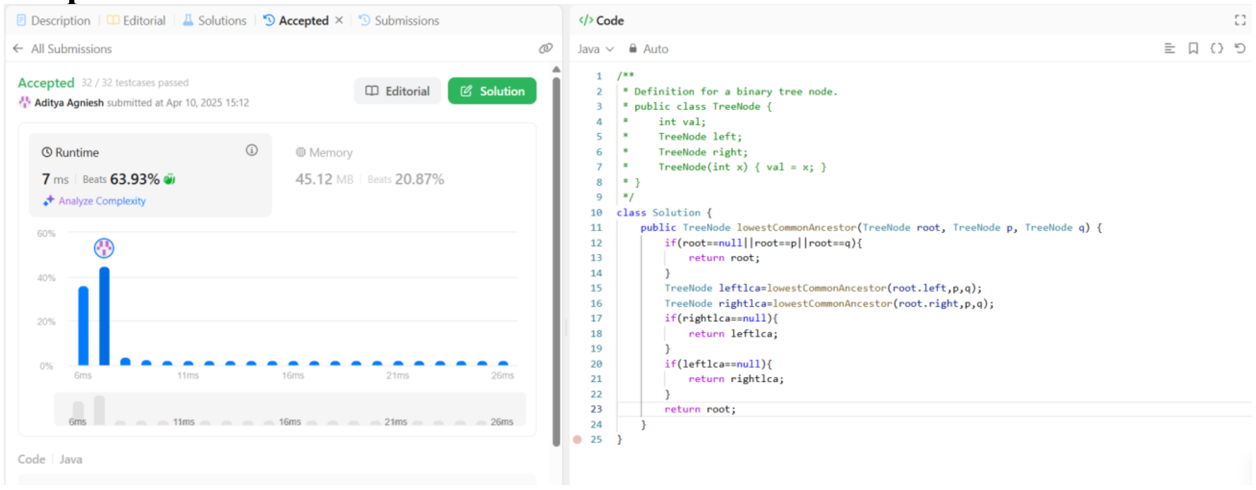
Ln 23, Col 10 Saved

Ques 4: Lowest Common Ancestor of a Binary Tree

Code:

```
class Solution {
    public TreeNode lowestCommonAncestor(TreeNode root, TreeNode p, TreeNode
q) {
        if(root==null||root==p||root==q){
            return root;
        }
        TreeNode leftlca=lowestCommonAncestor(root.left,p,q);
        TreeNode rightlca=lowestCommonAncestor(root.right,p,q);
        if(rightlca==null){
            return leftlca;
        }
        if(leftlca==null){
            return rightlca;
        }
        return root;
    }
}
```

Output:



Accepted 32 / 32 testcases passed
Aditya Agniesh submitted at Apr 10, 2025 15:12

Runtime 7 ms | Beats 63.93%
Memory 45.12 MB | Beats 20.87%

Code | Java

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode(int x) { val = x; }
 * }
 */
class Solution {
    public TreeNode lowestCommonAncestor(TreeNode root, TreeNode p, TreeNode q) {
        if(root==null||root==p||root==q){
            return root;
        }
        TreeNode leftlca=lowestCommonAncestor(root.left,p,q);
        TreeNode rightlca=lowestCommonAncestor(root.right,p,q);
        if(rightlca==null){
            return leftlca;
        }
        if(leftlca==null){
            return rightlca;
        }
        return root;
    }
}
```