

DSA Assignment - 8 July

Name: Aditya Tiwari

Roll No: 23/11/EC/040

LeetCode ID: adityatiwari1305t

GeeksForGeeks ID: aditya1305t

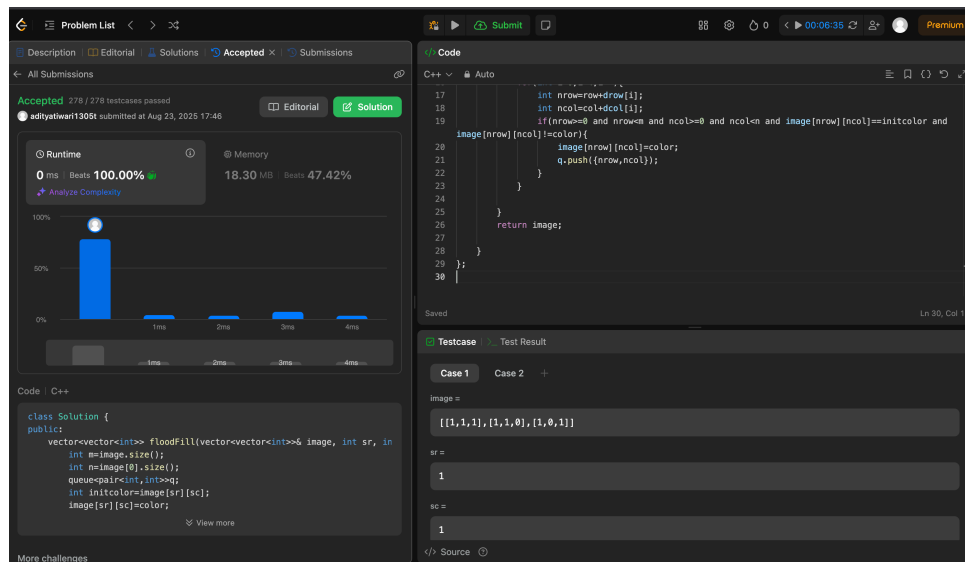
Github Repo Link:

https://github.com/Aditya1305T/SOE_Training_25

Question 1: Flood Fill

Platform: LeetCode

Link: - <https://leetcode.com/problems/flood-fill/description/>



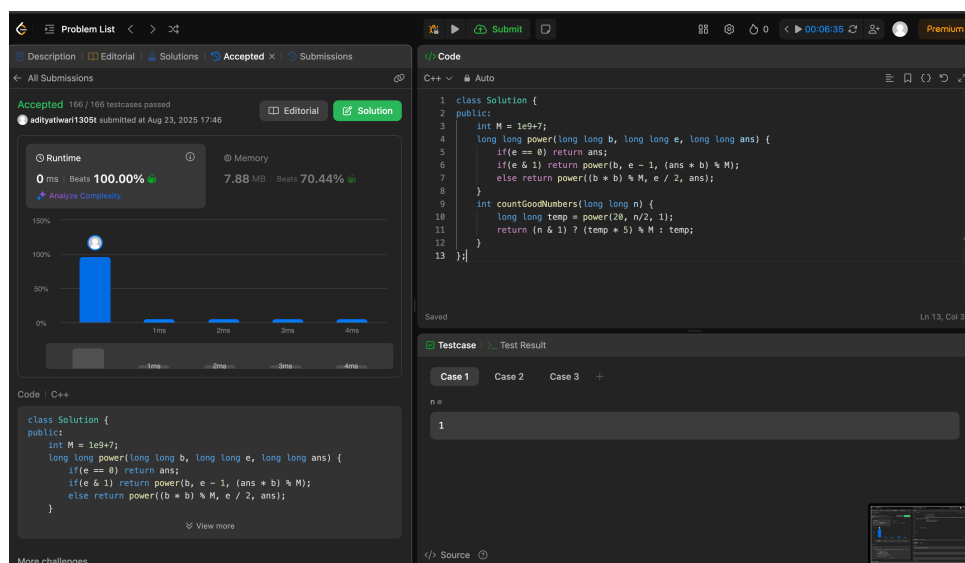
The screenshot shows the LeetCode interface for the 'Flood Fill' problem. The 'Accepted' status is confirmed with 278/278 test cases passed. The runtime is 0 ms (100.00% beats) and memory is 18.30 MB (47.42% beats). A bar chart shows the runtime performance relative to other solutions. The code is written in C++ and implements a recursive flood fill algorithm. The test case shows an input image of [[1,1,1],[1,1,0],[1,0,1]] with sr=1 and sc=1, resulting in the output image [[1,1,1],[1,1,0],[1,0,1]].

```
class Solution {
public:
    vector<vector<int>> floodFill(vector<vector<int>>& image, int sr, int sc, int color) {
        int n=image.size();
        int m=image[0].size();
        queue<pair<int,int>> q;
        int initcolor=image[sr][sc];
        image[sr][sc]=color;
        if (initcolor != color) {
            q.push({sr, sc});
            while (!q.empty()) {
                int row=q.front().first;
                int col=q.front().second;
                q.pop();
                if (row-1 >= 0 and row+1 < n and col-1 >= 0 and col+1 < m and image[row][col]==initcolor and image[row][col]!=color) {
                    image[row][col]=color;
                    q.push({row,col});
                }
            }
        }
        return image;
    }
};
```

Question 2: Count Good Numbers

Platform: LeetCode

Link: - <https://leetcode.com/problems/count-good-numbers/description>



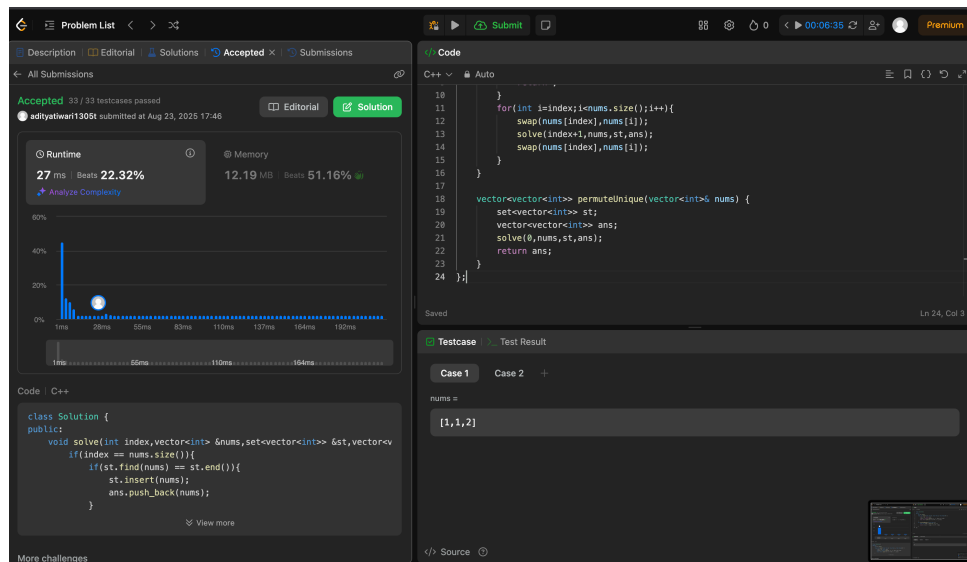
The screenshot shows the LeetCode interface for the 'Count Good Numbers' problem. The 'Accepted' status is confirmed with 166/166 test cases passed. The runtime is 0 ms (100.00% beats) and memory is 7.88 MB (70.44% beats). A bar chart shows the runtime performance relative to other solutions. The code is written in C++ and implements a recursive solution to count good numbers. The test case shows an input n=1, resulting in the output 1.

```
class Solution {
public:
    int M = 1e9+7;
    long long power(long long b, long long e, long long ans) {
        if(e == 0) return ans;
        if(e & 1) return power(b, e - 1, (ans * b) % M);
        else return power((b * b) % M, e / 2, ans);
    }
    int countGoodNumbers(long long n) {
        long long temp = power(20, n/2, 1);
        return (n & 1) ? (temp * 5) % M : temp;
    }
};
```

Question 3: Permutation II

Platform: LeetCode

Link: - <https://leetcode.com/problems/permutations-ii/description/>



Question 4: Rabbits in Forest

Platform: LeetCode

Link: - <https://leetcode.com/problems/rabbits-in-forest/description/>

