Daa Practical 8

Dipansh Tiwari

A4_B3_35

Aim: Implement Graph Colouring algorithm using the Graph colouring concept.

Code:

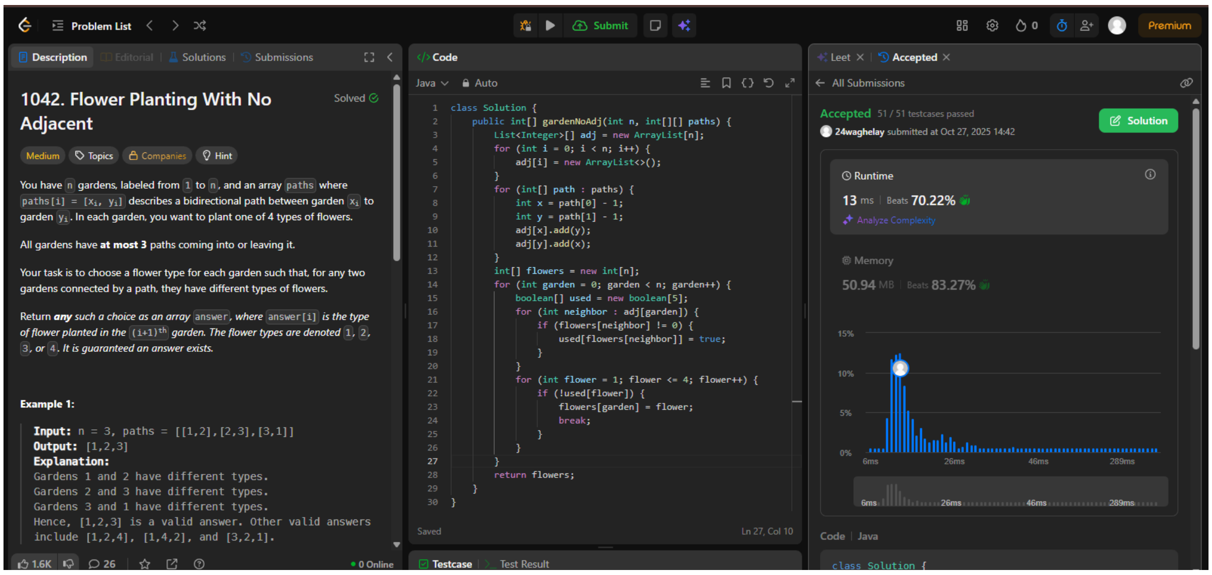
```
public class GraphColoring {
    final int V = 5;
    void graphColoring(int[][] graph) {
        int[] result = new int[V];
        result[0] = 0;
        for (int i = 1; i < V; i++)
            result[i] = -1;
        boolean[] available = new boolean[V];
        for (int u = 1; u < V; u++) {
            for (int i = 0; i < V; i++)
                if (graph[u][i] == 1 && result[i] != -1)
                    available[result[i]] = true;
            int cr;
            for (cr = 0; cr < V; cr++)
                if (!available[cr])
                    break;
            result[u] = cr;
            for (int i = 0; i < V; i++)
                available[i] = false;
        }
        for (int u = 0; u < V; u++)
            System.out.println("Vertex " + u + " --->  Color " + result[u]);
    }
    public static void main(String[] args) {
        GraphColoring g = new GraphColoring();
        int[][] graph = {
            {0, 1, 1, 1, 0},
            {1, 0, 1, 0, 0},
            {1, 1, 0, 1, 1},
            {1, 0, 1, 0, 1},
            {0, 0, 1, 1, 0}
        };
        g.graphColoring(graph);
    }
}
```

Output:

```
Vertex 0 ---> Color 0
Vertex 1 ---> Color 1
Vertex 2 ---> Color 2
Vertex 3 ---> Color 1
Vertex 4 ---> Color 0

=== Code Execution Successful ===
```

Leetcode Submission-



1042. Flower Planting With No Adjacent

Solved

Medium | Topics | Companies | Hint

You have `n` gardens, labeled from `1` to `n`, and an array `paths` where `paths[i] = [xi, yi]` describes a bidirectional path between garden `xi` to garden `yi`. In each garden, you want to plant one of 4 types of flowers.

All gardens have **at most 3** paths coming into or leaving it.

Your task is to choose a flower type for each garden such that, for any two gardens connected by a path, they have different types of flowers.

*Return **any** such a choice as an array `answer`, where `answer[i]` is the type of flower planted in the `(i+1)th` garden. The flower types are denoted `1`, `2`, `3`, or `4`. It is guaranteed an answer exists.*

Example 1:

```
Input: n = 3, paths = [[1,2],[2,3],[3,1]]
Output: [1,2,3]
Explanation:
Gardens 1 and 2 have different types.
Gardens 2 and 3 have different types.
Gardens 3 and 1 have different types.
Hence, [1,2,3] is a valid answer. Other valid answers include [1,2,4], [1,4,2], and [3,2,1].
```

```java
class Solution {
    public int[] gardenNoAdj(int n, int[][] paths) {
        List<Integer>[] adj = new ArrayList[n];
        for (int i = 0; i < n; i++) {
            adj[i] = new ArrayList<>();
        }
        for (int[] path : paths) {
            int x = path[0] - 1;
            int y = path[1] - 1;
            adj[x].add(y);
            adj[y].add(x);
        }
        int[] flowers = new int[n];
        for (int garden = 0; garden < n; garden++) {
            boolean[] used = new boolean[5];
            for (int neighbor : adj[garden]) {
                if (flowers[neighbor] != 0) {
                    used[flowers[neighbor]] = true;
                }
            }
            for (int flower = 1; flower <= 4; flower++) {
                if (!used[flower]) {
                    flowers[garden] = flower;
                    break;
                }
            }
        }
        return flowers;
    }
}
```

Saved   Ln 27, Col 10

Accepted 51 / 51 testcases passed

24waghelay submitted at Oct 27, 2025 14:42

Runtime: 13 ms | Beats 70.22%

Memory: 50.94 MB | Beats 83.27%