Problem 256: Paint House

Problem Information

Difficulty: Medium
Acceptance Rate: 64.05%
Paid Only: Yes
Tags: Array, Dynamic Programming

Problem Description

There is a row of `n` houses, where each house can be painted one of three colors: red, blue, or green. The cost of painting each house with a certain color is different. You have to paint all the houses such that no two adjacent houses have the same color.

The cost of painting each house with a certain color is represented by an `n x 3` cost matrix `costs`.

* For example, `costs[0][0]` is the cost of painting house `0` with the color red; `costs[1][2]` is the cost of painting house 1 with color green, and so on...

Return _the minimum cost to paint all houses_.

Example 1:

Input: costs = [[17,2,17],[16,16,5],[14,3,19]] **Output:** 10 **Explanation:** Paint house 0 into blue, paint house 1 into green, paint house 2 into blue. Minimum cost: 2 + 5 + 3 = 10.

Example 2:

Input: costs = [[7,6,2]] **Output:** 2

Constraints:

* `costs.length == n` * `costs[i].length == 3` * `1 <= n <= 100` * `1 <= costs[i][j] <= 20`

Code Snippets

C++:

```
class Solution {
public:
int minCost(vector<vector<int>>& costs) {

}
};
```

Java:

```
class Solution {
public int minCost(int[][] costs) {

}
}
```

Python3:

```
class Solution:
def minCost(self, costs: List[List[int]]) -> int:
```