

题目介绍^[1]

假如有一排房子，共 n 个，每个房子可以被粉刷成红色、蓝色或者绿色这三种颜色中的一种，你需要粉刷所有的房子并且使其相邻的两个房子颜色不能相同。

当然，因为市场上不同颜色油漆的价格不同，所以房子粉刷成不同颜色的花费成本也是不同的。每个房子粉刷成不同颜色的花费是以一个 $[n \times 3]$ 的正整数矩阵 `costs` 来表示的。

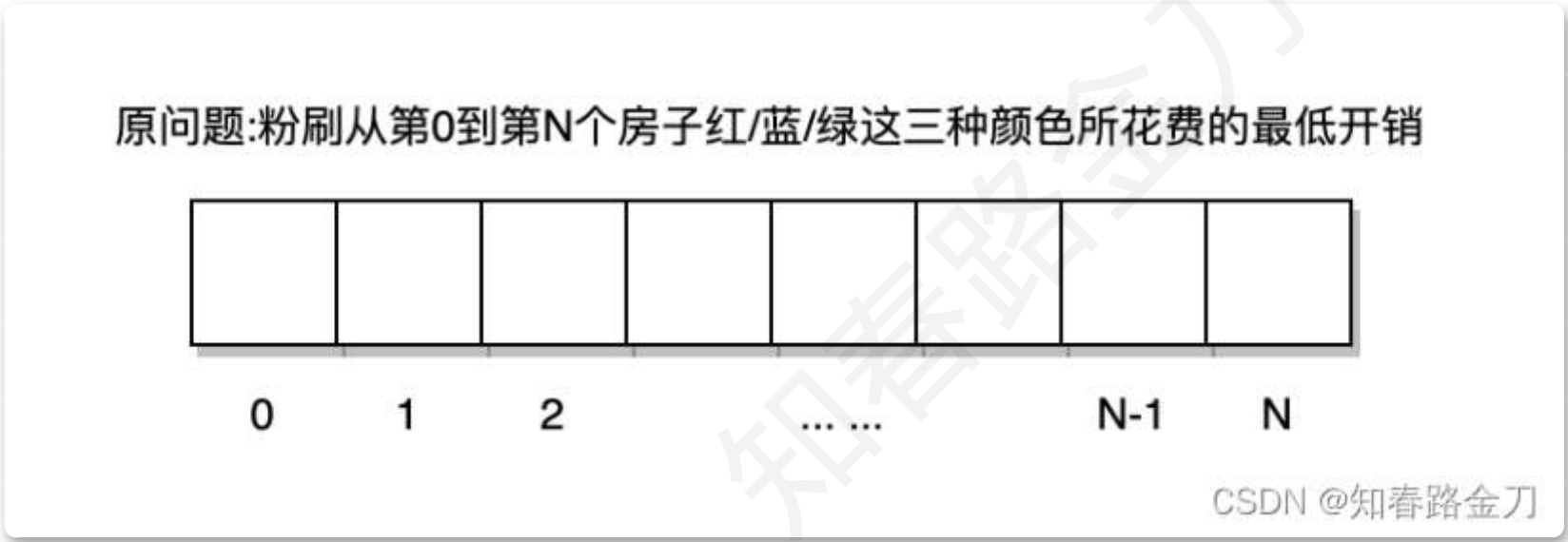
例如，`costs[0][0]` 表示第 0 号房子粉刷成红色的成本花费；`costs[1][2]` 表示第 1 号房子粉刷成绿色的花费，以此类推。

请计算出粉刷完所有房子最少的花费成本。

题目解答

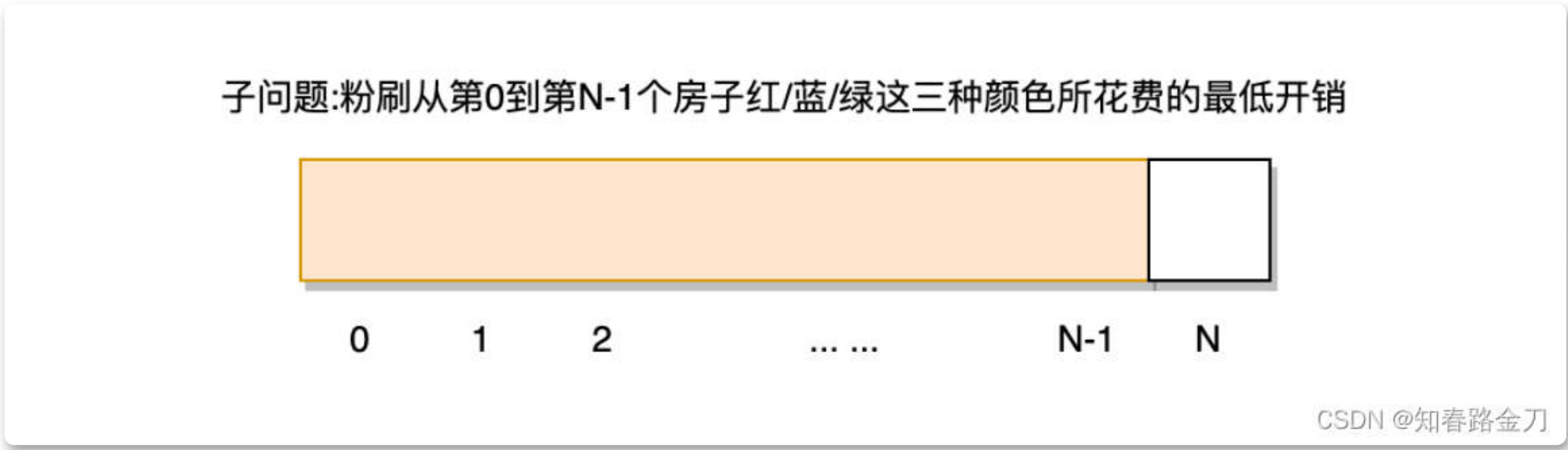
又又又又是动态规划，动态规划的要点是啥来着？发现子问题、找出状态转换方程、优化数组空间。

首先寻找子问题



题目的原问题是求解「粉刷从第 0 到第 N 个房子红/蓝/绿这三种颜色所花费的最低开销」，这个问题可以拆成如下 N 个子问题

- 粉刷第 0 个房子红/蓝/绿这三种颜色所花费的最低开销
- 粉刷从第 0 到第 1 个房子红/蓝/绿这三种颜色所花费的最低开销
-
- 粉刷从第 0 到第 $N-2$ 个房子红/蓝/绿这三种颜色所花费的最低开销
- 粉刷从第 0 到第 $N-1$ 个房子红/蓝/绿这三种颜色所花费的最低开销



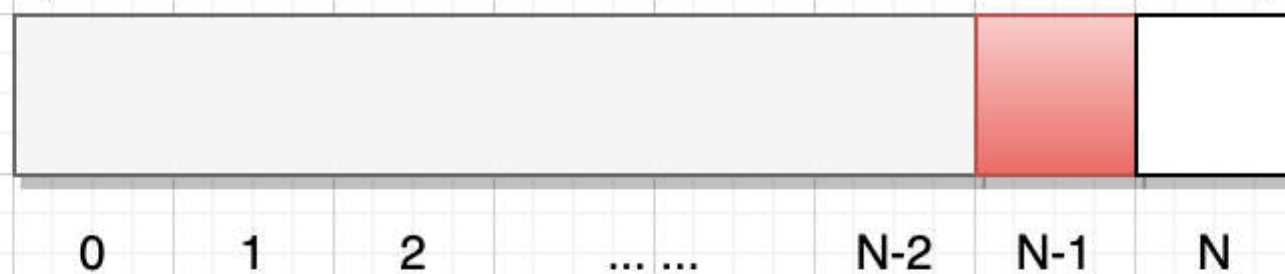
注意这题有一个限制就是「相邻的两个房子颜色不能相同」，所以小粉刷匠每刷一个房子的时候，他需要思考这个房子要刷哪种颜色，刷红色？刷蓝色？刷绿色？这样每一个子问题又可以继续拆解变成如下 $3N$ 个子问题



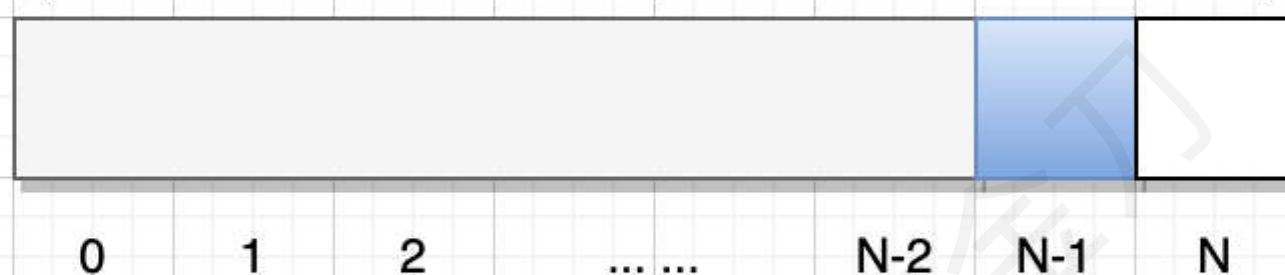
微信搜一搜

知春路金刀

子问题1:粉刷从第0到第N-1个房子红/蓝/绿这三种颜色所花费的最低开销
(其中第N-2个房子不能是红色, 因为第N-1个房子要刷红色)



子问题2:粉刷从第0到第N-1个房子红/蓝/绿这三种颜色所花费的最低开销
(其中第N-2个房子不能是蓝色, 因为第N-1个房子要刷蓝色)



子问题3:粉刷从第0到第N-1个房子红/蓝/绿这三种颜色所花费的最低开销
(其中第N-2个房子不能是绿色, 因为第N-1个房子要刷绿色)



CSDN @知春路金刀

- 粉刷第 0 个房子红/蓝/绿这三种颜色所花费的最低开销
 - 给第 0 个房子刷红色时, 粉刷从第 0 到第 0 个房子的最低开销是多少?
 - 给第 0 个房子刷蓝色时, 粉刷从第 0 到第 0 个房子的最低开销是多少?
 - 给第 0 个房子刷绿色时, 粉刷从第 0 到第 0 个房子的最低开销是多少?
- 粉刷从第 0 到第 1 个房子红/蓝/绿这三种颜色所花费的最低开销
 - 给第 1 个房子刷红色时, 粉刷从第 0 到第 1 个房子的最低开销是多少?
 - 给第 1 个房子刷蓝色时, 粉刷从第 0 到第 1 个房子的最低开销是多少?
 - 给第 1 个房子刷绿色时, 粉刷从第 0 到第 1 个房子的最低开销是多少?
-
- 粉刷从第 0 到第 N-2 个房子红/蓝/绿这三种颜色所花费的最低开销
 - 给第 N-2 个房子刷红色时, 粉刷从第 0 到第 N-2 个房子的最低开销是多少?
 - 给第 N-2 个房子刷蓝色时, 粉刷从第 0 到第 N-2 个房子的最低开销是多少?
 - 给第 N-2 个房子刷绿色时, 粉刷从第 0 到第 N-2 个房子的最低开销是多少?



微信搜一搜

知春路金刀

- 粉刷从第 0 到第 N-1 个房子红/蓝/绿这三种颜色所花费的最低开销
 - 给第 N-1 个房子刷红色时，粉刷从第 0 到第 N-1 个房子的最低开销是多少？
 - 给第 N-1 个房子刷蓝色时，粉刷从第 0 到第 N-1 个房子的最低开销是多少？
 - 给第 N-1 个房子刷绿色时，粉刷从第 0 到第 N-1 个房子的最低开销是多少？



确定状态转移方程

子问题已经确定出来了，那么如果我们知道了「粉刷从第 0 到第 N-1 个房子红/蓝/绿这三种颜色所花费的最低开销」，那么我们如何根据这个子问题来算出原问题「粉刷从第 0 到第 N 个房子红/蓝/绿这三种颜色所花费的最低开销」呢？

粉刷匠为了让开销达到最小，自学了编程然后搞了 3 个数组 red、blue 和 green，red[k]表示「粉刷从第 0 到第 k 个房子红/蓝/绿这三种颜色所花费的最低开销，其中第 k 个房子粉刷为红色」，blue[k]和 green[k]亦然，粉刷匠每到达一个房子的时候，都会去更新 red[k]、blue[k]和 green[k]，当粉刷匠来到了第 K 个房子的时心里可能这么想：

- 我要把第 K 个房子刷为红色
 - 粉刷匠决定把第 K 个房子刷为红色，并记录下当前第 K 个房子刷为红色时所花费的最低开销为 $\text{red}[k] = \min(\text{blue}[k-1], \text{green}[k-1]) + \text{cost}[k]$ 。
 - 「解释」：既然粉刷匠想把第 K 个房子刷为红色，且想保持所花费的开销是最低的，那第 K-1 个房子不能是红色的且要选择 blue[k-1]和 green[k-1]这两个数其中最小的一个。
- 我要把第 K 个房子刷为蓝色
 - 同上
- 我要把第 K 个房子刷为绿色
 - 同上

那么我们得出了状态转移方程如下

```
red[k] = min(blue[k-1], green[k-1]) + cost[k]
blue[k] = min(red[k-1], green[k-1]) + cost[k]
green[k] = min(blue[k-1], red[k-1]) + cost[k]
```

有了状态转移方程，那就很容易写出代码了



微信搜一搜

知春路金刀

代码实现(一维动态规划)

```
class Solution {
    public int minCost(int[][] costs) {
        int[][] dp = new int[costs.length][3];
        // 0 - 红色
        // 1 - 蓝色
        // 2 - 绿色
        dp[0][0] = costs[0][0];
        dp[0][1] = costs[0][1];
        dp[0][2] = costs[0][2];
        for (int i = 1; i < costs.length; i++) {
            dp[i][0] = Math.min(dp[i-1][1], dp[i-1][2]) + costs[i][0];
            dp[i][1] = Math.min(dp[i-1][0], dp[i-1][2]) + costs[i][1];
            dp[i][2] = Math.min(dp[i-1][0], dp[i-1][1]) + costs[i][2];
        }
        return Math.min(dp[costs.length-1][0], Math.min(dp[costs.length-1][1], dp[costs.length-1][2]));
    }
}
```

复杂度分析

- 时间复杂度： $O(n)$
- 空间复杂度： $O(n)$

代码实现(动态规划优化)

上面的一维动态规划解法使用了一个 `dp` 数组，我们仔细观察可以发现，计算 `dp[i]` 的状态只取决于 `dp[i-1]` 的状态，所以我们可以用三个临时变量 `red/blue/green` 来代替 `dp[i-1][0]/dp[i-1][1]/dp[i-1][2]` 中的值。

```
class Solution {
    public int minCost(int[][] costs) {
        int[][] dp = new int[costs.length][3];
        int redCost = costs[0][0], blueCost = costs[0][1], greenCost = costs[0][2];
        for (int i = 1; i < costs.length; i++) {
            int newRedCost = Math.min(blueCost, greenCost) + costs[i][0];
            int newBlueCost = Math.min(redCost, greenCost) + costs[i][1];
            int newGreenCost = Math.min(redCost, blueCost) + costs[i][2];
            redCost = newRedCost;
            blueCost = newBlueCost;
            greenCost = newGreenCost;
        }
        return Math.min(redCost, Math.min(blueCost, greenCost));
    }
}
```

复杂度分析

- 时间复杂度： $O(n)$
- 空间复杂度： $O(1)$

参考资料

- [1] 原题链接：
<https://leetcode-cn.com/problems/paint-house/>



微信搜一搜

知春路金刀