



链滴

# [每日 LeetCode] 746. Min Cost Climbing Stairs

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1555162728999>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Description:

On a staircase, the  $i$ -th step has some non-negative cost  $cost[i]$  assigned (0 indexed).

Once you pay the cost, you can either climb one or two steps. You need to find minimum cost to reach the top of the floor, and you can either start from the step with index 0, or the step with index 1.

### Example 1:

Input:  $cost = [10, 15, 20]$

Output: 15

Explanation: Cheapest is start on  $cost[1]$ , pay that cost and go to the top.

### Example 2:

Input:  $cost = [1, 100, 1, 1, 1, 100, 1, 1, 100, 1]$

Output: 6

Explanation: Cheapest is start on  $cost[0]$ , and only step on 1s, skipping  $cost[3]$ .

### Note:

1.  $cost$  will have a length in the range  $[2, 1000]$ .
2. Every  $cost[i]$  will be an integer in the range  $[0, 999]$ .

---

思路：本题题意是有一个楼梯，要离开 $i$ 层需要付 $cost[i]$ 的费用，每次可以爬1层或2层，问最少花多少能够达到顶楼。考虑动态规划思想，找到动态规划三要素（问题的阶段，每个阶段的状态，从前一个阶段转化到后一个阶段之间的递推关系）求解即可。

---

### C++代码

```
class Solution {
public:
    int minCostClimbingStairs(vector<int>& cost) {
        int n = cost.size();
        vector<int> dp(n, 0);
        dp[0] = cost[0];
        dp[1] = cost[1];
        for (int i = 2; i < n; ++i)
            dp[i] = min(dp[i - 1], dp[i - 2]) + cost[i];
        return min(dp[n - 1], dp[n - 2]);
    }
};
```

---

运行时间: 8ms

运行内存: 8.9M