746 Min Cost Climbing Stairs

2018年5月4日 15:03

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].

来自 <https://leetcode.com/problems/min-cost-climbing-stairs/description/>

数组的每个索引做为一个阶梯，第 i个阶梯对应着一个非负数的体力花费值 cost[i](索引从0开始)。
每当你爬上一个阶梯你都要花费对应的体力花费值，然后你可以选择继续爬一个阶梯或者爬两个阶梯。
您需要找到达到楼层顶部的最低花费。在开始时，你可以选择从索引为 0 或 1 的元素作为初始阶梯。

示例 1:

输入: cost = [10, 15, 20]

输出: 15

解释: 最低花费是从cost[1]开始，然后走两步即可到阶梯顶，一共花费15。

示例 2:

输入: cost = [1, 100, 1, 1, 1, 100, 1, 1, 100, 1]

输出: 6

解释: 最低花费方式是从cost[0]开始，逐个经过那些1，跳过cost[3]，一共花费6。

注意:

1. cost 的长度将会在 [2, 1000]。
2. 每一个 cost[i] 将会是一个Integer类型，范围为 [0, 999]。

Solution for Python3:

```python
class Solution:
    def minCostClimbingStairs(self, cost):
        """
        :type cost: List[int]
        :rtype: int
        """
        f1 = f2 = 0
        for x in reverse(cost):
            f1, f2 = x + min(f1, f2), f1
        return min(f1, f2)
```

Solution for C++:

```cpp
class Solution {
public:
    int minCostClimbingStairs(vector<int>&
cost) {
        int f1 = 0, f2 = 0, tmp;
        for (int i = cost.size() - 1; i >= 0;
i--) {
            tmp = f1;
            f1 = cost[i] + min(f1, f2);
            f2 = tmp;
        }
        return min(f1, f2);
    }
};
```