

746. Min Cost Climbing Stairs

Description (/problems/min-cost-climbing-stairs/description/)

Hints (/problems/min-cost-climbing-stairs/hints/)

Submission

□ Pick One (/problems/random-one-question/)

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:

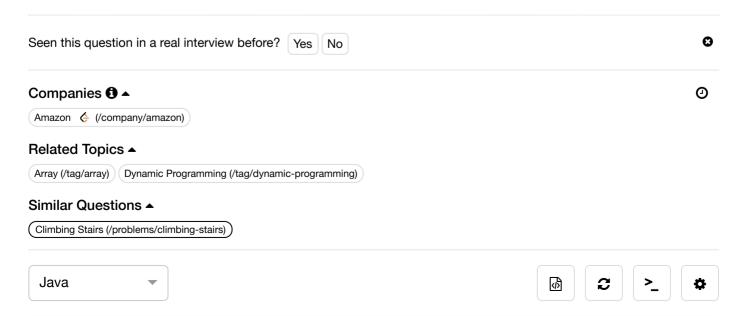
Input: cost = [1, 100, 1, 1, 1, 100, 1, 1, 100, 1]

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].



```
1 v class Solution {
        public int minCostClimbingStairs(int[] cost) {
 2 •
            int cost2 = cost[cost.length - 1], cost1 = cost[cost.length - 2];
 3 ▼
            for (int i=cost.length - 3; i \ge 0; i = 0) {
 4 ▼
 5 ▼
                 int cur = cost[i] + Math.min(cost2, cost1);
                 cost2 = cost1;
6
7
                 cost1 = cur;
8
9
            return Math.min(cost2, cost1);
10
        }
11
   }
```

☐ Custom Testcase (Contribute ●)

▶ Run
♠ Submit

Copyright © 2018 LeetCode

Contact Us (/support/) | Frequently Asked Questions (/faq/) | Terms of Service (/terms/) | Privacy Policy (/privacy/)

States (/region/)