746. Min Cost Climbing Stairs
Easy

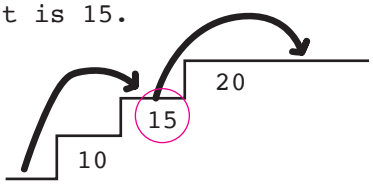
You are given an integer array cost where `cost[i]` is the cost of `i`th step on a staircase.
Once you pay the cost, you can either climb one or two steps.

You can either start from the step with index 0, or the step with index 1.

Return the minimum cost to reach the top of the floor.

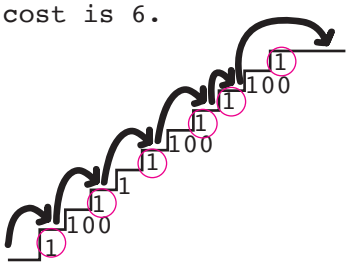
```
Example 1:

Input: cost = [10,15,20]
Output: 15
Explanation: You will start at index 1.
- Pay 15 and climb two steps to reach the top.
The total cost is 15.
```




```
Example 2:

Input: cost = [1,100,1,1,1,100,1,1,100,1]
Output: 6
Explanation: You will start at index 0.
- Pay 1 and climb two steps to reach index 2.
- Pay 1 and climb two steps to reach index 4.
- Pay 1 and climb two steps to reach index 6.
- Pay 1 and climb one step to reach index 7.
- Pay 1 and climb two steps to reach index 9.
- Pay 1 and climb one step to reach the top.
The total cost is 6.
```




```python
import collections
class Solution:
    def minCostClimbingStairs(self, cost: List[int]) -> int:
        for i in range(len(cost) - 3, -1, -1):
            cost[i] += min(cost[i + 1], cost[i + 2])

        return min(cost[0], cost[1])
```

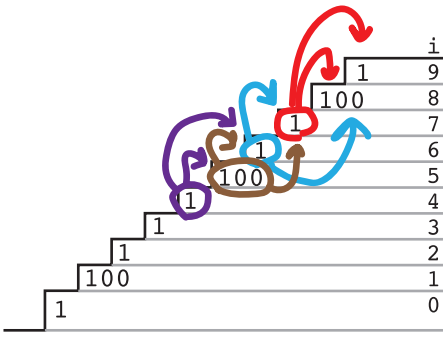
```
example 2:
cost = [1,100,1,1,1,100,1,1,100,1]

    len(cost) -> 10
    for i in range(7, -1, -1)
i = 7 { cost[i] = min(cost[i+1], cost[i+2])
      [1,100,1,1,1,100,1,2,100,1]

i = 6 { cost[i] = min(cost[i+1], cost[i+2])
      [1,100,1,1,1,100,3,2,100,1]

i = 5 { cost[i] = min(cost[i+1], cost[i+2])
      [1,100,1,1,1,102,3,2,100,1]

i = 5 { cost[i] = min(cost[i+1], cost[i+2])
      [1,100,1,1,4,102,3,2,100,1]
```




```
cost[0] because it records the smallest (minimum)
    between cost[i+1], cost[i+2] is the smallest possible value
```