746. Min Cost Climbing Stairs

You are given an integer array `cost` where `cost[i]` is the cost of `ith` 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: 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].

Constraints:

- `2 <= cost.length <= 1000`
- `0 <= cost[i] <= 999`
- ```
  def minCostClimbingStairs(self, cost: List[int]) -> int:
          n = len(cost)
          dp = [0]*(n)
          dp[n-1] = cost[n-1]
          dp[n-2] = cost[n-2]
          for i in range(n-3,-1,-1):
              dp[i] = cost[i]+min(dp[i+1],dp[i+2])
          return min(dp[0],dp[1])
  ```
  ```