Problem 3502: Minimum Cost to Reach Every Position

Problem Information

Difficulty: Easy
Acceptance Rate: 82.86%
Paid Only: No
Tags: Array

Problem Description

You are given an integer array `cost` of size `n`. You are currently at position `n` (at the end of the line) in a line of `n + 1` people (numbered from 0 to `n`).

You wish to move forward in the line, but each person in front of you charges a specific amount to **swap** places. The cost to swap with person `i` is given by `cost[i]`.

You are allowed to swap places with people as follows:

* If they are in front of you, you **must** pay them `cost[i]` to swap with them. * If they are behind you, they can swap with you for free.

Return an array `answer` of size `n`, where `answer[i]` is the **minimum** total cost to reach each position `i` in the line.

Example 1:

Input: cost = [5,3,4,1,3,2]

Output: [5,3,3,1,1,1]

Explanation:

We can get to each position in the following way:

* `i = 0`. We can swap with person 0 for a cost of 5. * `i = 1`. We can swap with person 1 for a cost of 3. * `i = 2`. We can swap with person 1 for a cost of 3, then swap with person 2 for free. * `i = 3`. We can swap with person 3 for a cost of 1. * `i = 4`. We can swap with person 3 for a cost of 1, then swap with person 4 for free. * `i = 5`. We can swap with person 3 for a cost of 1, then swap with person 5 for free.

Example 2:

Input: cost = [1,2,4,6,7]

Output: [1,1,1,1,1]

Explanation:

We can swap with person 0 for a cost of 1, then we will be able to reach any position `i` for free.

Constraints:

* `1 <= n == cost.length <= 100` * `1 <= cost[i] <= 100`

Code Snippets

C++:

```
class Solution {
public:
vector<int> minCosts(vector<int>& cost) {

}
};
```

Java:

```
class Solution {
public int[] minCosts(int[] cost) {

}
}
```

Python3:

```
class Solution:
def minCosts(self, cost: List[int]) -> List[int]:
```