Problem 2361: Minimum Costs Using the Train Line

Problem Information

Difficulty: Hard
Acceptance Rate: 77.79%
Paid Only: Yes
Tags: Array, Dynamic Programming

Problem Description

A train line going through a city has two routes, the regular route and the express route. Both routes go through the **same** `n + 1` stops labeled from `0` to `n`. Initially, you start on the regular route at stop `0`.

You are given two **1-indexed** integer arrays `regular` and `express`, both of length `n`. `regular[i]` describes the cost it takes to go from stop `i - 1` to stop `i` using the regular route, and `express[i]` describes the cost it takes to go from stop `i - 1` to stop `i` using the express route.

You are also given an integer `expressCost` which represents the cost to transfer from the regular route to the express route.

Note that:

* There is no cost to transfer from the express route back to the regular route. * You pay `expressCost` **every** time you transfer from the regular route to the express route. * There is no extra cost to stay on the express route.

Return _a**1-indexed** array _`costs` _of length_`n` _, where_`costs[i]`_is the**minimum** cost to reach stop _`i` _from stop_`0`.

Note that a stop can be counted as **reached** from either route.

Example 1:

Input: regular = [1,6,9,5], express = [5,2,3,10], expressCost = 8 **Output:** [1,7,14,19] **Explanation:** The diagram above shows how to reach stop 4 from stop 0 with minimum cost. - Take the regular route from stop 0 to stop 1, costing 1. - Take the express route from stop 1 to stop 2, costing 8 + 2 = 10. - Take the express route from stop 2 to stop 3, costing 3. - Take the regular route from stop 3 to stop 4, costing 5. The total cost is 1 + 10 + 3 + 5 = 19. Note that a different route could be taken to reach the other stops with minimum cost.

Example 2:

Input: regular = [11,5,13], express = [7,10,6], expressCost = 3 **Output:** [10,15,24] **Explanation:** The diagram above shows how to reach stop 3 from stop 0 with minimum cost. - Take the express route from stop 0 to stop 1, costing 3 + 7 = 10. - Take the regular route from stop 1 to stop 2, costing 5. - Take the express route from stop 2 to stop 3, costing 3 + 6 = 9. The total cost is 10 + 5 + 9 = 24. Note that the expressCost is paid again to transfer back to the express route.

Constraints:

* `n == regular.length == express.length` * `1 <= n <= 105` * `1 <= regular[i], express[i], expressCost <= 105`

Code Snippets

C++:

```
class Solution {
public:
vector<long long> minimumCosts(vector<int>& regular, vector<int>& express,
int expressCost) {

}
};
```

Java:

```
class Solution {
public long[] minimumCosts(int[] regular, int[] express, int expressCost) {

}
}
```

Python3:

```
class Solution:
def minimumCosts(self, regular: List[int], express: List[int], expressCost:
int) -> List[int]:
```