

2969. Minimum Number of Coins for Fruits II Premium

Hard Topics Hint

You are at a fruit market with different types of exotic fruits on display.

You are given a **1-indexed** array `prices`, where `prices[i]` denotes the number of coins needed to purchase the i^{th} fruit.

The fruit market has the following offer:

- If you purchase the i^{th} fruit at `prices[i]` coins, you can get the next i fruits for free.

Note that even if you **can** take fruit j for free, you can still purchase it for `prices[j]` coins to receive a new offer.

Return *the **minimum** number of coins needed to acquire all the fruits.*

Example 1:

Input: `prices = [3,1,2]`
Output: `4`
Explanation: You can acquire the fruits as follows:
– Purchase the 1st fruit with 3 coins, and you are allowed to take the 2nd fruit for free.
– Purchase the 2nd fruit with 1 coin, and you are allowed to take the 3rd fruit for free.
– Take the 3rd fruit for free.
Note that even though you were allowed to take the 2nd fruit for free, you purchased it because it is more optimal.
It can be proven that 4 is the minimum number of coins needed to acquire all the fruits.

Example 2:

Input: `prices = [1,10,1,1]`
Output: `2`
Explanation: You can acquire the fruits as follows:
– Purchase the 1st fruit with 1 coin, and you are allowed to take the 2nd fruit for free.
– Take the 2nd fruit for free.
– Purchase the 3rd fruit for 1 coin, and you are allowed to take the 4th fruit for free.
– Take the 4th fruit for free.
It can be proven that 2 is the minimum number of coins needed to acquire all the fruits.

Constraints:

- `1 <= prices.length <= 105`
- `1 <= prices[i] <= 105`

Seen this question in a real interview before? 1/5

Yes No

Accepted 666 | Submissions 1.4K | Acceptance Rate 49.0%

Topics

ArrayDynamic ProgrammingQueueHeap (Priority Queue)Monotonic Queue

Hint 1

Use DP + Fenwick tree.

Hint 2

There is another creative solution using multiset instead of Fenwick.

Hint 3

Imagine we want to calculate `dp[i]` which is the answer to the problem for the first i fruits.

Hint 4

If we buy i^{th} fruit from the set of indices: `[(i + 1) / 2, (i + 1) / 2 + 1, (i + 1) / 2 + 2, ..., i - 1]`, then we can get fruits `l + 1, l + 2, ..., i` for free.

Hint 5

We just need to get all the first `l - 1` fruits as well and the minimum price for that, is `dp[l - 1]`.

Hint 6

So at the index i , we are looking for such an index l that `dp[l - 1] + prices[l]` is as minimum as possible.

Hint 7

We can store these values in a multiset and update the values in it.

Discussion (2)