Problem 2944: Minimum Number of Coins for Fruits

Problem Information

Difficulty: Medium
Acceptance Rate: 48.08%
Paid Only: No
Tags: Array, Dynamic Programming, Queue, Heap (Priority Queue), Monotonic Queue

Problem Description

You are given an **0-indexed** integer array `prices` where `prices[i]` denotes the number of coins needed to purchase the `(i + 1)th` fruit.

The fruit market has the following reward for each fruit:

* If you purchase the `(i + 1)th` fruit at `prices[i]` coins, you can get any number of 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 - 1]` coins to receive its reward.

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

Example 1:

Input: prices = [3,1,2]

Output: 4

Explanation:

* Purchase the 1st fruit with `prices[0] = 3` coins, you are allowed to take the 2nd fruit for free.
* Purchase the 2nd fruit with `prices[1] = 1` coin, you are allowed to take the 3rd fruit for free. * Take the 3rd fruit for free.

Note that even though you could take the 2nd fruit for free as a reward of buying 1st fruit, you purchase it to receive its reward, which is more optimal.

Example 2:

Input: prices = [1,10,1,1]

Output: 2

Explanation:

* Purchase the 1st fruit with `prices[0] = 1` coin, you are allowed to take the 2nd fruit for free. * Take the 2nd fruit for free. * Purchase the 3rd fruit for `prices[2] = 1` coin, you are allowed to take the 4th fruit for free. * Take the 4th fruit for free.

Example 3:

Input: prices = [26,18,6,12,49,7,45,45]

Output: 39

Explanation:

* Purchase the 1st fruit with `prices[0] = 26` coin, you are allowed to take the 2nd fruit for free. * Take the 2nd fruit for free. * Purchase the 3rd fruit for `prices[2] = 6` coin, you are allowed to take the 4th, 5th and 6th (the next three) fruits for free. * Take the 4th fruit for free. * Take the 5th fruit for free. * Purchase the 6th fruit with `prices[5] = 7` coin, you are allowed to take the 8th and 9th fruit for free. * Take the 7th fruit for free. * Take the 8th fruit for free.

Note that even though you could take the 6th fruit for free as a reward of buying 3rd fruit, you purchase it to receive its reward, which is more optimal.

Constraints:

* `1 <= prices.length <= 1000` * `1 <= prices[i] <= 105`

Code Snippets

C++:

```
class Solution {
public:
int minimumCoins(vector<int>& prices) {

}
};
```

Java:

```
class Solution {
public int minimumCoins(int[] prices) {

}
}
```

Python3:

```
class Solution:
def minimumCoins(self, prices: List[int]) -> int:
```