Problem 3592: Inverse Coin Change

Problem Information

Difficulty: Medium
Acceptance Rate: 50.78%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

You are given a **1-indexed** integer array `numWays`, where `numWays[i]` represents the number of ways to select a total amount `i` using an **infinite** supply of some _fixed_ coin denominations. Each denomination is a **positive** integer with value **at most** `numWays.length`.

However, the exact coin denominations have been _lost_. Your task is to recover the set of denominations that could have resulted in the given `numWays` array.

Return a **sorted** array containing **unique** integers which represents this set of denominations.

If no such set exists, return an **empty** array.

Example 1:

Input: numWays = [0,1,0,2,0,3,0,4,0,5]

Output: [2,4,6]

Explanation:

Amount | Number of ways | Explanation ---|---|--- 1 | 0 | There is no way to select coins with total value 1. 2 | 1 | The only way is `[2]`. 3 | 0 | There is no way to select coins with total value 3. 4 | 2 | The ways are `[2, 2]` and `[4]`. 5 | 0 | There is no way to select coins with total value 5. 6 | 3 | The ways are `[2, 2, 2]`, `[2, 4]`, and `[6]`. 7 | 0 | There is no way to select coins with total value 7. 8 | 4 | The ways are `[2, 2, 2, 2]`, `[2, 2, 4]`, `[2, 6]`, and `[4, 4]`. 9 | 0 | There is no way to select coins with total value 9. 10 | 5 | The ways are `[2, 2, 2, 2, 2]`, `[2, 2, 2, 4]`, `[2, 4, 4]`, `[2, 2, 6]`, and `[4, 6]`. **Example 2:**

Input: numWays = [1,2,2,3,4]

Output: [1,2,5]

Explanation:

Amount | Number of ways | Explanation ---|---|--- 1 | 1 | The only way is `[1]`. 2 | 2 | The ways are `[1, 1]` and `[2]`. 3 | 2 | The ways are `[1, 1, 1]` and `[1, 2]`. 4 | 3 | The ways are `[1, 1, 1, 1]`, `[1, 1, 2]`, and `[2, 2]`. 5 | 4 | The ways are `[1, 1, 1, 1, 1]`, `[1, 1, 1, 2]`, `[1, 2, 2]`, and `[5]`. **Example 3:**

Input: numWays = [1,2,3,4,15]

Output: []

Explanation:

No set of denomination satisfies this array.

Constraints:

* `1 <= numWays.length <= 100` * `0 <= numWays[i] <= 2 * 108`

Code Snippets

C++:

```
class Solution {
public:
vector<int> findCoins(vector<int>& numWays) {

}
};
```

Java:

```
class Solution {
public List<Integer> findCoins(int[] numWays) {

}
}
```

Python3:

```
class Solution:
def findCoins(self, numWays: List[int]) -> List[int]:
```