6355. Prime Subtraction Operation

My Submissions (/contest/weekly-contest-338/problems/prime-subtraction-operation/submissions/) Back to Contest (/contest/weekly-contest-338/)

User Accepted:	166
User Tried:	458
Total Accepted:	170
Total Submissions:	605
Difficulty:	Medium

You are given a **0-indexed** integer array `nums` of length `n`.

You can perform the following operation as many times as you want:

- Pick an index `i` that you haven't picked before, and pick a prime `p` **strictly less than** `nums[i]`, then subtract `p` from `nums[i]`.

Return *true if you can make `nums` a strictly increasing array using the above operation and false otherwise.*

A **strictly increasing array** is an array whose each element is strictly greater than its preceding element.

Example 1:

```
Input: nums = [4,9,6,10]
Output: true
Explanation: In the first operation: Pick i = 0 and p = 3, and then subtract 3 from nums[0], so that nums becomes [1,9,6,10].
In the second operation: i = 1, p = 7, subtract 7 from nums[1], so nums becomes equal to [1,2,6,10].
After the second operation, nums is sorted in strictly increasing order, so the answer is true.
```

Example 2:

```
Input: nums = [6,8,11,12]
Output: true
Explanation: Initially nums is sorted in strictly increasing order, so we don't need to make any operations.
```

Example 3:

```
Input: nums = [5,8,3]
Output: false
Explanation: It can be proven that there is no way to perform operations to make nums sorted in strictly increasing order, so t
```

Constraints:

- `1 <= nums.length <= 1000`
- `1 <= nums[i] <= 1000`
- `nums.length == n`

JavaScript

```
const primeSubOperation = (a) => {
    let se = sieveEratosthenes(1005), pre = 0;
    for (const x of a) {
        let cur = x;
        for (const p of se) {
            if (p <= x && x - p > pre) cur = Math.min(cur, x - p);
        }
        if (cur <= pre) return false;
        pre = cur;
    }
    return true;
};

const sieveEratosthenes = (n) => {
    let prime = Array(n + 1).fill(true), res = new Set();
    for (let p = 2; p * p <= n; p++) {
        if (prime[p] == true) {
            for (let i = p * p; i <= n; i += p) prime[i] = false;
        }
    }
    for (let p = 2; p <= n; p++) {
        if (prime[p]) res.add(p);
    }
    return res;
};
```

□ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/923557577/) More Details > (/submissions/detail/923557577/)

Share your acceptance!

6

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)