2523. Closest Prime Numbers in Range

My Submissions (/contest/weekly-contest-326/problems/closest-prime-numbers-in-range/submissions/) Back to Contest (/contest/weekly-contest-326/)

Given two positive integers `left` and `right`, find the two integers `num1` and `num2` such that:

- `left <= nums1 < nums2 <= right` .
- `nums1` and `nums2` are both **prime** numbers.
- `nums2 - nums1` is the **minimum** amongst all other pairs satisfying the above conditions.

Return *the positive integer array* `ans = [nums1, nums2]` . *If there are multiple pairs satisfying these conditions, return the one with the minimum* `nums1` *value or* `[-1, -1]` *if such numbers do not exist.*

A number greater than `1` is called **prime** if it is only divisible by `1` and itself.

User Accepted:	3543
User Tried:	4852
Total Accepted:	3784
Total Submissions:	13251
Difficulty:	Medium

Example 1:

```
Input: left = 10, right = 19
Output: [11,13]
Explanation: The prime numbers between 10 and 19 are 11, 13, 17, and 19.
The closest gap between any pair is 2, which can be achieved by [11,13] or [17,19].
Since 11 is smaller than 17, we return the first pair.
```

Example 2:

```
Input: left = 4, right = 6
Output: [-1,-1]
Explanation: There exists only one prime number in the given range, so the conditions cannot be satisfied.
```

Constraints:

- $1 <= left <= right <= 10^6$

Discuss (https://leetcode.com/problems/closest-prime-numbers-in-range/discuss)

JavaScript

```javascript
const closestPrimes = (l, r) => {
    let a = sieveEratosthenes(r), idx = -1, res = [-1, -1], dis = Number.MAX_SAFE_INTEGER;
    a = [...a].sort((x, y) => x - y);
    for (let i = 0; i < a.length; i++) {
        if (a[i] >= l) {
            idx = i;
            break;
        }
    }
    if (idx == -1) return res;
    a = a.slice(idx);
    for (let i = 1; i < a.length; i++) {
        if (a[i] - a[i - 1] < dis) {
            dis = a[i] - a[i - 1];
            res = [a[i - 1], a[i]];
        }
    }
    return res;
};

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

```
    }
    let res = new Set();
    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/871281955/) | More Details > (/submissions/detail/871281955/)

Share your acceptance!