6115. Count the Number of Ideal Arrays

My Submissions (/contest/weekly-contest-301/problems/count-the-number-of-ideal-arrays/submissions/) Back to Contest (/contest/weekly-contest-301/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

You are given two integers `n` and `maxValue`, which are used to describe an **ideal** array.

A **0-indexed** integer array `arr` of length `n` is considered **ideal** if the following conditions hold:

- Every `arr[i]` is a value from `1` to `maxValue`, for `0 <= i < n`.
- Every `arr[i]` is divisible by `arr[i - 1]`, for `0 < i < n`.

Return *the number of **distinct** ideal arrays of length* `n`. Since the answer may be very large, return it modulo $10^9 + 7$.

Example 1:

```
Input: n = 2, maxValue = 5
Output: 10
Explanation: The following are the possible ideal arrays:
- Arrays starting with the value 1 (5 arrays): [1,1], [1,2], [1,3], [1,4], [1,5]
- Arrays starting with the value 2 (2 arrays): [2,2], [2,4]
- Arrays starting with the value 3 (1 array): [3,3]
- Arrays starting with the value 4 (1 array): [4,4]
- Arrays starting with the value 5 (1 array): [5,5]
There are a total of 5 + 2 + 1 + 1 + 1 = 10 distinct ideal arrays.
```

Example 2:

```
Input: n = 5, maxValue = 3
Output: 11
Explanation: The following are the possible ideal arrays:
- Arrays starting with the value 1 (9 arrays):
   - With no other distinct values (1 array): [1,1,1,1,1]
   - With 2nd distinct value 2 (4 arrays): [1,1,1,1,2], [1,1,1,2,2], [1,1,2,2,2], [1,2,2,2,2]
   - With 2nd distinct value 3 (4 arrays): [1,1,1,1,3], [1,1,1,3,3], [1,1,3,3,3], [1,3,3,3,3]
- Arrays starting with the value 2 (1 array): [2,2,2,2,2]
- Arrays starting with the value 3 (1 array): [3,3,3,3,3]
There are a total of 9 + 1 + 1 = 11 distinct ideal arrays.
```

Constraints:

- $2 <= n <= 10^4$
- $1 <= maxValue <= 10^4$

JavaScript

```
const ll = BigInt, mod = ll(1e9 + 7), N = 1e4 + 15;

const hcomb = (p, q) => p == 0 && q == 0 ? 1 : comb(p + q - 1, q);
const comb_init = () => {
    fact[0] = ifact[0] = inv[1] = 1n;
    for (let i = 2; i < N; i++) inv[i] = (mod - mod / ll(i)) * inv[mod % ll(i)] % mod;
    for (let i = 1; i < N; i++) {
        fact[i] = fact[i - 1] * ll(i) % mod;
        ifact[i] = ifact[i - 1] * inv[i] % mod;
    }
};

// combination mod pick k from n
const comb = (n, k) => {
    if (n < k || k < 0) return 0;
    return fact[n] * ifact[k] % mod * ifact[n - k] % mod;
};

const number_factor = (n) => {
    let m = new Map();
    for (let i = 2; i * i <= n; i++) {
        while (n % i == 0) {
```

```
            n /= i;
            m.set(i, m.get(i) + 1 || 1);
        }
    }
    if (n > 1) m.set(n, m.get(n) + 1 || 1);
    return m;
};

let fact, ifact, inv;
const idealArrays = (n, maxValue) => {
    fact = Array(N).fill(0), ifact = Array(N).fill(0), inv = Array(N).fill(0);
    comb_init();
    let res = 0n;
    for (let x = 1; x <= maxValue; x++) {
        let perm = 1n, m = number_factor(x);
        for (const [x, occ] of m) {
            perm = perm * hcomb(n, occ) % mod;
        }
        res += perm;
    }
    return res % mod;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/743235485/) | More Details > (/submissions/detail/743235485/)

Share your acceptance!