2514. Count Anagrams

My Submissions (/contest/biweekly-contest-94/problems/count-anagrams/submissions/) Back to Contest (/contest/biweekly-contest-94/)

User Accepted:	1239
User Tried:	2641
Total Accepted:	1292
Total Submissions:	6020
Difficulty:	Hard

You are given a string `s` containing one or more words. Every consecutive pair of words is separated by a single space `' '`.

A string `t` is an **anagram** of string `s` if the i^{th} word of `t` is a **permutation** of the i^{th} word of `s`.

- For example, `"acb dfe"` is an anagram of `"abc def"`, but `"def cab"` and `"adc bef"` are not.

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

Example 1:

```
Input: s = "too hot"
Output: 18
Explanation: Some of the anagrams of the given string are "too hot", "oot hot", "oto toh", "too toh", and "too oht".
```

Example 2:

```
Input: s = "aa"
Output: 1
Explanation: There is only one anagram possible for the given string.
```

Constraints:

- $1 <= s.length <= 10^5$
- `s` consists of lowercase English letters and spaces `' '`.
- There is single space between consecutive words.

Discuss (https://leetcode.com/problems/count-anagrams/discuss)

JavaScript

```
const ord = (c) => c.charCodeAt();

const ll = BigInt, mod = ll(1e9 + 7), N = 1e5 + 5;

let fact, ifact, inv;
const comb_init = () => {
    fact = Array(N), ifact = Array(N), inv = Array(N);
    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;
    }
};

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

const countAnagrams = (ss) => {
    let a = ss.split(" "), res = 1n;
    comb_init();
    for (const s of a) {
        let f = Array(26).fill(0), len = s.length;
        for (const c of s) f[ord(c) - 97]++;
        res = res * fact[len] % mod;
        for (const occ of f) res = res * ifact[occ] % mod;
    }
    return res;
```

```
31 };
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/865216522/) | More Details > (/submissions/detail/865216522/)

Share your acceptance!