**1220. Count Vowels Permutation: -**

Hard Accepted: 135.2K Submissions: 220K Acceptance Rate: 61.5%

Given an integer n, your task is to count how many strings of length n can be formed under the following rules:

* Each character is a lower case vowel ('a', 'e', 'i', 'o', 'u')
* Each vowel 'a' may only be followed by an 'e'.
* Each vowel 'e' may only be followed by an 'a' or an 'i'.
* Each vowel 'i' **may not** be followed by another 'i'.
* Each vowel 'o' may only be followed by an 'i' or a 'u'.
* Each vowel 'u' may only be followed by an 'a'.

Since the answer may be too large, return it modulo 10^9 + 7.

**Example 1:**

**Input:** n = 1

**Output:** 5

**Explanation:** All possible strings are: "a", "e", "i" , "o" and "u".

**Example 2:**

**Input:** n = 2

**Output:** 10

**Explanation:** All possible strings are: "ae", "ea", "ei", "ia", "ie", "io", "iu", "oi", "ou" and "ua".

**Example 3:**

**Input:** n = 5

**Output:** 68

**Constraints:**

* 1 <= n <= 2 \* 10^4

**Code: -**

int ind(char ch){

    if(ch == 'a')   return 0;

    if(ch == 'e')   return 1;

    if(ch == 'i')   return 2;

    if(ch == 'o')   return 3;

    return 4;

}

vector<char> v = {'a', 'e', 'i', 'o', 'u'};

int mod = 1e9 + 7;

class Solution {

public:

    long long helper(int start, char ch, int n, vector<vector<long long>> &dp){

        // base case

        if(start == n - 1)      return 1;

        // dp found case

        if(dp[start][ind(ch)] != -1)

            return dp[start][ind(ch)];

        // recursive case

        long long count = 0;

        if(ch == 'a')

            count = (count + helper(start+1, 'e', n, dp)) % mod;

        else if(ch == 'e'){

            count = (count + helper(start+1, 'a', n, dp)) % mod;

            count = (count + helper(start+1, 'i', n, dp)) % mod;

        }

        else if(ch == 'i'){

            count = (count + helper(start+1, 'a', n, dp)) % mod;

            count = (count + helper(start+1, 'e', n, dp)) % mod;

            count = (count + helper(start+1, 'o', n, dp)) % mod;

            count = (count + helper(start+1, 'u', n, dp)) % mod;

        }

        else if(ch == 'o'){

            count = (count + helper(start+1, 'i', n, dp)) % mod;

            count = (count + helper(start+1, 'u', n, dp)) % mod;

        }

        else if(ch == 'u')

            count = (count + helper(start+1, 'a', n, dp)) % mod;

        // return from current state

        return dp[start][ind(ch)] = count;

    }

    int countVowelPermutation(int n) {

        vector<vector<long long>> dp(n, vector<long long>(5, -1));

        long long count = 0;

        for(char &i : v){

            count = (count + helper(0, i, n, dp)) % mod;

        }

        return count;

    }

};

**T.C: - O(N \* 5)**

**S.C: - O(N \* 5)**