1641. Count Sorted Vowel Strings

Given an integer `n`, return *the number of strings of length* `n` *that consist only of vowels (*`a`*,* `e`*,* `i`*,* `o`*,* `u`*) and are* ***lexicographically sorted****.*

A string `s` is **lexicographically sorted** if for all valid `i`, `s[i]` is the same as or comes before `s[i+1]` in the alphabet.

Example 1:

Input: n = 1
Output: 5
Explanation: The 5 sorted strings that consist of vowels only are `["a","e","i","o","u"].`

Example 2:

Input: n = 2
Output: 15
Explanation: The 15 sorted strings that consist of vowels only are
["aa","ae","ai","ao","au","ee","ei","eo","eu","ii","io","iu","oo","ou","uu"].
Note that "ea" is not a valid string since 'e' comes after 'a' in the alphabet.

Example 3:

Input: n = 33
Output: 66045

Constraints:

- `1 <= n <= 50`

```
def countVowelStrings(self, n: int) -> int:
        if n==1:
            return 5
        dp = [5,4,3,2,1]
        if n==2:
            return sum(dp)
        for i in range(3,n+1):
            j = 0
            while j<5:
                dp[j] = sum(dp[j:])
                j = j+1
        return sum(dp)
```

Approach 2: Backtracking

```
def countVowelStrings(self, n: int) -> int:
        res = []
        ref = ['a','e','i','o','u']
        for i in ref:
            self.permute(i,n,res)
        return len(res)


    def permute(self,ssf,n,res):
        if len(ssf)==n:
            res.append(ssf)
            return

        for i in ['a','e','i','o','u']:
            if ssf[-1]<=i:
                self.permute(ssf+i,n,res)
```