1. Substring with Concatenation of All Words

Hard

You are given a string, **s**, and a list of words, **words**, that are all of the same length. Find all starting indices of substring(s) in **s** that is a concatenation of each word in **words**exactly once and without any intervening characters.

**Example 1:**

Input:  
 s = "barfoothefoobarman",  
 words = ["foo","bar"]  
Output: [0,9]  
Explanation: Substrings starting at index 0 and 9 are "barfoor" and "foobar" respectively.  
The output order does not matter, returning [9,0] is fine too.

**Example 2:**

Input:  
 s = "wordgoodgoodgoodbestword",  
 words = ["word","good","best","word"]  
Output: []

**解法1**

先求出words的排列，然后用bm算法进行匹配。但是会超时

**解法2**

考虑到条件中words中的字符串长度相等的条件，使用滑动窗口

class Solution {  
public:  
 vector<int> findSubstring(string s, vector<string>& words) {  
 vector<int>ans;  
 int n = words.size();  
 if(n == 0 || s.size() == 0)return ans;  
 int len = words[0].size();  
 for(int i = 0; i <= (int)s.size() - n \* len; ++i){  
 unordered\_map<string, int>mp;  
 for(string str : words)mp[str]++;  
 int j;  
 for(j = 0; j < n; ++j){  
 string temp = s.substr(i + j \* len, len);  
 if(find(words.begin(), words.end(), temp) != words.end() && mp[temp] > 0)mp[temp]--;  
 else break;  
 }  
 bool flag = true;  
 for(string str : words){  
 if(mp[str] != 0){  
 flag = false;  
 break;  
 }  
 }  
 if(flag)ans.push\_back(i);  
 }  
 return ans;  
 }  
};