567. Permutation in String

Given two strings `s1` and `s2`, return `true` if `s2` contains a permutation of `s1`, or `false` otherwise.

In other words, return `true` if one of `s1`'s permutations is the substring of `s2`.

Example 1:

```
Input: s1 = "ab", s2 = "eidbaooo"
Output: true
Explanation: s2 contains one permutation of s1 ("ba").
```

Example 2:

```
Input: s1 = "ab", s2 = "eidboaoo"
Output: false
```

Constraints:

- `1 <= s1.length, s2.length <= 10`^{4}
- `s1` and `s2` consist of lowercase English letters.

567. Permutation in String

```
/*
    substring and count
    Time complexity: O(26+m+n(m+26+26))=O(nm)
    space complexity: O(26+n*(m+26))=O(nm)
*/

class Solution {
    public:
        bool checkInclusion(std::string s1,std::string s2){
            int n=s2.size();
            int m=s1.size();

            if(n<m) return false;

            std::vector<int> freq1(26,0);
            for(auto& c: s1) freq1[c-'a']++;

            for(int i=0;i<=n-m;++i){
                std::string sub=s2.substr(i,m);
                std::vector<int> freq(26,0);
                for(auto& c: sub) freq[c-'a']++;
                if(freq==freq1) return true;
            }
            return false;
        }
};
```

567. Permutation in String

```
/*
    Counting Sliding window
    Time complexity: O(26+m+26+m+m+26+n*(26+26))=O(n+m)
    space complexity: O(26+26+m)=O(m)
*/
class Solution {
    public:
        bool checkInclusion(std::string s1,std::string s2){
            int n=s2.size();
            int m=s1.size();

            if(n<m) return false;

            std::vector<int> freq1(26,0);
            for(auto& c: s1) freq1[c-'a']++;

            // Create window
            std::vector<int> window(26,0);
            std::string sub=s2.substr(0,m);
            for(auto& c: sub) window[c-'a']++;

            if(window==freq1) return true;

            // Slide the window
            for(int i=1;i<=n-m;++i){
                for(int j=0;j<26;++j){
                    window[j]-=int((s2[i-1]-'a')==j);
                    window[j]+=int((s2[i+m-1]-'a')==j);
                }
                if(window==freq1) return true;
            }
            return false;
        }
};
```