

→ Permutation in Strings

S1 = "ab"

S2 = "eidbaooo"

↓
permutations

exist in
of not

exists - true
Not exists - false

In permutations

[No. of characters + frequency of characters]

↓
remain same

① Store the frequency of characters in S1.

0	1	2	3	4
a	b	c	d	e

24	25
24	25

```
int freq[26] = {0};  
freq[S1[i] - 'a']++
```

② Search S1 permutation in S2.
Window Based.

S1 = ab

e	i	d	b	a	o	o	o
---	---	---	---	---	---	---	---

K K K

↓
true

otherwise

false


```

bool isfreqsame(freq1[], freq2[]) {
    for (i = 0; i < 26; i++) {
        if (freq1[i] != freq2[i]) {
            return false;
        }
    }
    return true;
}

```

```

bool checkInclusion(string s1, s2) {
    int freq[26] = {0};
    for (int i = 0; i < s1.length(); i++) {
        freq[s1[i] - 'a']++;
    }
}

```

```

int windowSize = s1.length();
for (int i = 0; i < s2.length(); i++) {
    int windowIdx = 0, idx = i;
    windowFreq[26] = {0};
    while (windowIdx < windowSize ||
           idx < s2.length()) {
        windowFreq[s2[idx] - 'a']++;
        windowIdx++, idx++;
    }
    if (isfreqsame(freq, windowFreq))
        return true;
}
}

```