Leetcode May Challenge DAY: 18

1. **Python**

```python
class Solution:
  def checkInclusion(self, s1: str, s2: str) -> bool:
    if len(s1)==0 and len(s2)==0 :
      return False
    a=[0]*26
    for each in s1 :
      a[ord(each)-ord("a")]+=1
    i,j=0,len(s1)
    l=[]
    while j<len(s2)+1 :
      l=s2[i:j]
      t=[0]*26
      for each in l :
        t[ord(each)-ord("a")]+=1
      if a==t :
        return True
      t[ord(s2[i])-ord("a")]-=1
      i+=1
      j+=1
    return False
```

2. C++

```cpp
class Solution {
public:
    bool checkInclusion(string s1, string s2) {
        int n = s1.size(), m = s2.size(), ar[26] = {0}, currCount = n, start = 0;
        for(auto c : s1) ++ar[c-'a'];
        for(int i = 0;i<m;i++) {
            if (ar[s2[i]-'a']-- > 0) --currCount;
            while(currCount == 0) {
                if ((i - start + 1) == n) return true;
                if (ar[s2[start++] - 'a']++ == 0) ++currCount;
            }
        }
        return false;
    }
};
```

3. JAVA

```java
class Solution {

    public boolean checkInclusion(String s1, String s2) {

        int[] count = new int[128];

        for (char c: s1.toCharArray()) count[c]++;

        int start = 0, end = 0;

        while (end < s2.length()) {

            count[s2.charAt(end)]--;

            while (count[s2.charAt(end)] < 0) {

                count[s2.charAt(start++)]++;

            }

            if (end - start + 1 == s1.length()) return true;

            end++;

        }

        return false;

    }
}
```