1.Easy\07.Valid_anagram.cpp

```
/*
QUESTION:-
Given two strings s and t, return true if t is an anagram of s, and false otherwise.
An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

Example 1:

Input: s = "anagram", t = "nagaram"
Output: true
Example 2:

Input: s = "rat", t = "car"
Output: false

Problem: Valid Anagram

Approach:
1. Create an unordered map to store the count of each character in string `s`.
2. Iterate over each character in `s` and increment its count in the map.
3. Iterate over each character in `t`.
   - If the character is not present in the map or its count is zero, return false.
   - Decrement the count of the character in the map.
   - If the count becomes zero, remove the character from the map.
4. After iterating through all characters in `t`, if the map is empty, return true; otherwise, return false.

Code:
*/

bool isAnagram(string s, string t) {
    unordered_map<char,int>mp;
    for(auto c:s)
        mp[c]++;
    for(auto c:t){
        if(mp.find(c)==mp.end())
            return false;
        mp[c]--;
        if(mp[c]==0)
            mp.erase(c);
    }
    return (mp.size()==0);
}

/*
Time complexity: O(max(s.length(), t.length()))
Space complexity: O(s.length())
*/
```