242. Valid Anagram

Solved

Easy | Topics | Companies

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
```

Constraints:

- `1 <= s.length, t.length <= 5 * 10^4`
- `s` and `t` consist of lowercase English letters.

Valid Anagram >

A valid anagram is a word or a phrase that is formed by rearranging the letters of another word or phrase, using all the original letter exactly one.

Input : s = anagram t = nagram

output true.

(Approach 1) Sorting approach

(1) Check lengths

Both string are of equal length than proceed

(2) Convert Strings to character Array

s = ['a', 'n', 'a', 'g', 'r', 'a', 'm']

t = ['n', 'a', 'g', 'a', 'r', [redacted], 'm']

(3) Sort Both array Now

(4) Compare sorted Array.

Time complexity O(N Log N)

</> Code

ava Auto

```java
class Solution {
    public boolean isAnagram(String s, String t) {
        if (s.length() != t.length()) {
            return false;
        }
        char[] sArray = s.toCharArray();
        char[] tArray = t.toCharArray();
        Arrays.sort(sArray);
        Arrays.sort(tArray);
        return Arrays.equals(sArray, tArray);
    }
}
```

Saved Ln 10, Col 46

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

s =

"anagram"

t =

"nagaram"

(Approach 2) Fixed-Size Array Approach (for ASCII Characters)

① Check lengths
Both string are of equal length than proceed.

② Initialize Count Array
Create an array of size 26
The index of an array corresponds to each letter ('a' is index 0, 'b' is index 1, ... 'z' is index 25)

③ Traverse the characters of 's' and increment the count in the array of each character

④ Traverse the characters of 't' and decrement the count in the array of each character

}

(5) If In count Array values at each index(Pos) is 0 than return true Else return false

Code

Java | Auto

```java
public boolean isAnagram(String s, String t) {
    if (s.length() != t.length()) {
        return false;
    }

    int[] count = new int[26];

    for (int c=0;c<s.length();c++) {
        count[s.charAt(c) - 'a']++;
    }

    for (int c=0;c<t.length();c++) {
        count[t.charAt(c) - 'a']--;
    }

    for(int counts: count){
        if(counts != 0){
            return false;
        }

    }

    return true;
}
}
```

Saved

Ln 1, Col

Testcase | Test Result

Accepted Runtime: 0 ms

20. Valid Parentheses

Solved

Easy | Topics | Companies | Hint

Given a string `s` containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.
3. Every close bracket has a corresponding open bracket of the same type.

Example 1:

```
Input: s = "()"
Output: true
```

Example 2:

```
Input: s = "()[]{}"
Output: true
```

Example 3:

```
Input: s = "(]"
Output: false
```

Constraints:

Valid parenthesis

Hints.

1. Open Brackets must be closed by the same type of brackets.
2. Open bracket must be closed in the correct order
3. Every close bracket has a corresponding open bracket of the same type

Approach>

- Look for the most recent bracket
- Means you are looking for the last bracket you just saw
- last In - first Out = Stack Data Structure

```java
class Solution {
    public boolean isValid(String s) {

  Stack<Character> stack = new Stack<>();

        for (char c : s.toCharArray()) {
            if (c == '(') {
                stack.push(')');
            } else if (c == '{') {
                stack.push('}');
            } else if (c == '[') {
                stack.push(']');
            } else if (c == ')' || c == '}' || c == ']') {
                if (stack.isEmpty() || stack.pop() != c) {
                    return false;
                }
            } else {
                // Optionally handle invalid characters if needed
                continue;
            }
        }

        return stack.isEmpty();
    }
}
```

Accepted

Azan imtiaz submitted at Aug 02, 2024 12:38

Editorial | Solution

Runtime

1 ms | Beats 98.92%

Analyze Complexity

Memory

41.58 MB | Beats 30.19%

Code | Java

```java
class Solution {
    public boolean isValid(String s) {

  Stack<Character> stack = new Stack<>();

        for (char c : s.toCharArray()) {
            if (c == '(') {
                stack.push(')');
```