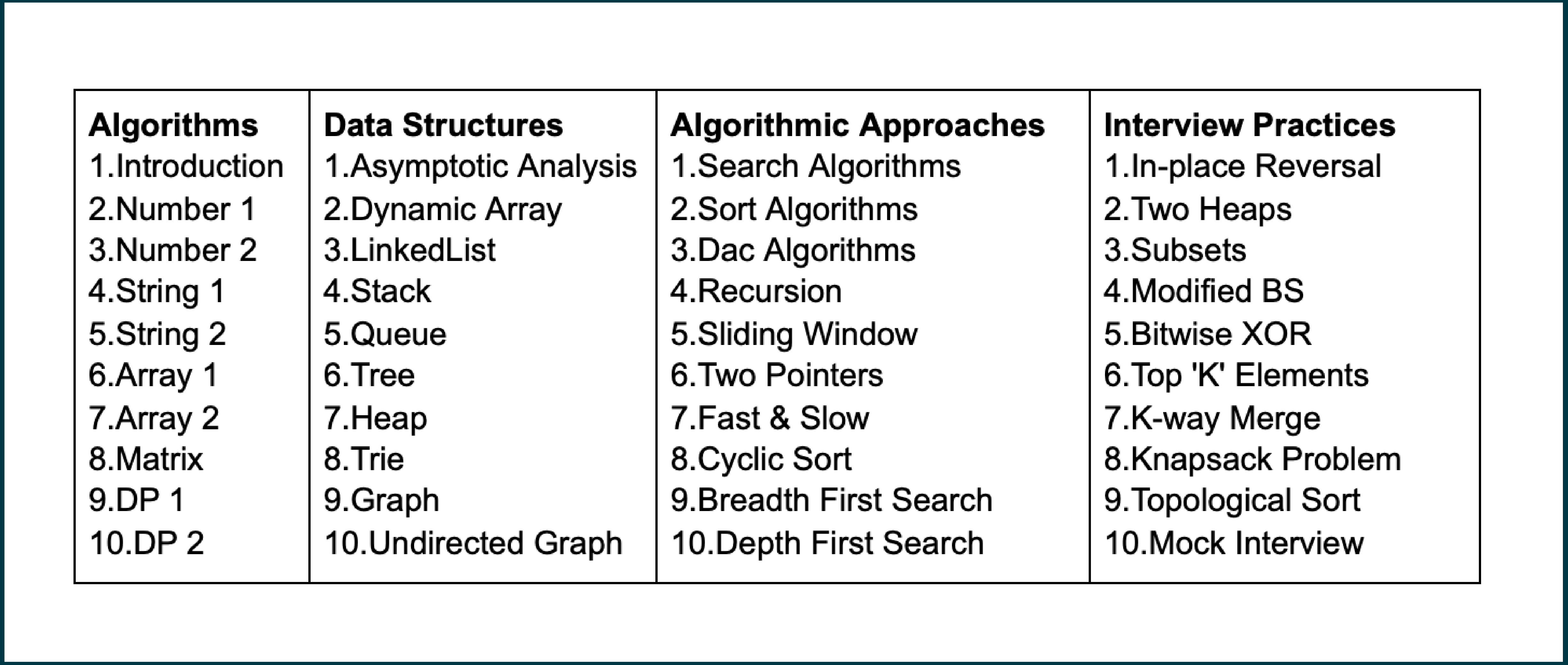
PDP ONLINE

DSA - Algorithms

String 1

Course Planning

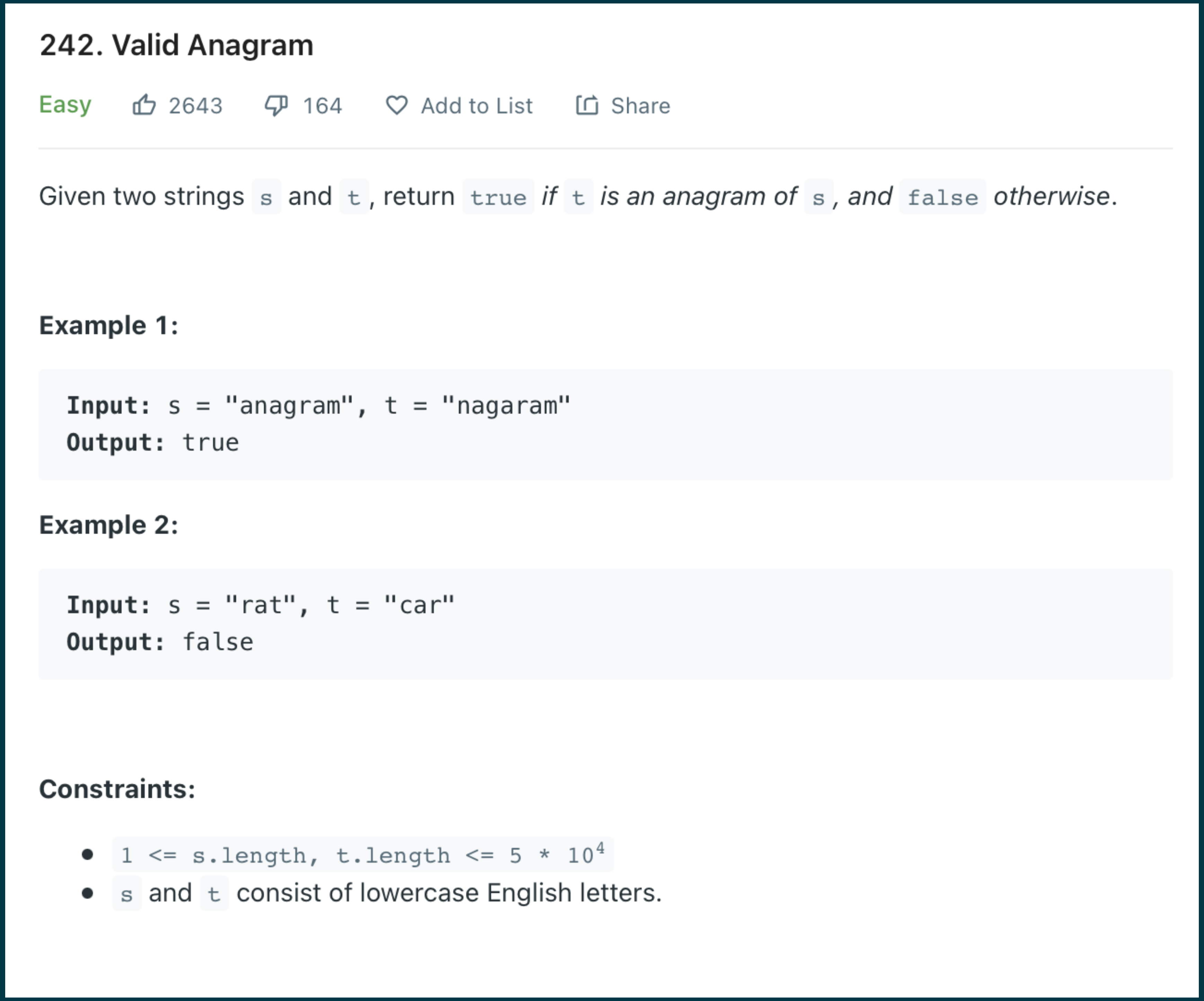


Algorithms	Data Structures	Algorithmic Approaches	Interview Practices
1.Introduction	1.Asymptotic Analysis	1.Search Algorithms	1.In-place Reversal
2.Number 1	2.Dynamic Array	2.Sort Algorithms	2.Two Heaps
3.Number 2	3.LinkedList	3.Dac Algorithms	3.Subsets
4.String 1	4.Stack	4.Recursion	4.Modified BS
5.String 2	5.Queue	5.Sliding Window	5.Bitwise XOR
6.Array 1	6.Tree	6.Two Pointers	6.Top 'K' Elements
7.Array 2	7.Heap	7.Fast & Slow	7.K-way Merge
8.Matrix	8.Trie	8.Cyclic Sort	8.Knapsack Problem
9.DP 1	9.Graph	9.Breadth First Search	9.Topological Sort
10.DP 2	10.Undirected Graph	10.Depth First Search	10.Mock Interview

https://kurbanovxurshid.medium.com/data-structures-and-algorithms-course-plan-1ab912ec1e17

Asked by Facebook

Explanation



242. Valid Anagram

Easy 2643 164 Add to List Share

Given two strings `s` and `t`, return `true` *if* `t` *is an anagram of* `s`*, and* `false` *otherwise*.

Example 1:

```
Input: s = "anagram", t = "nagaram"
Output: true
```

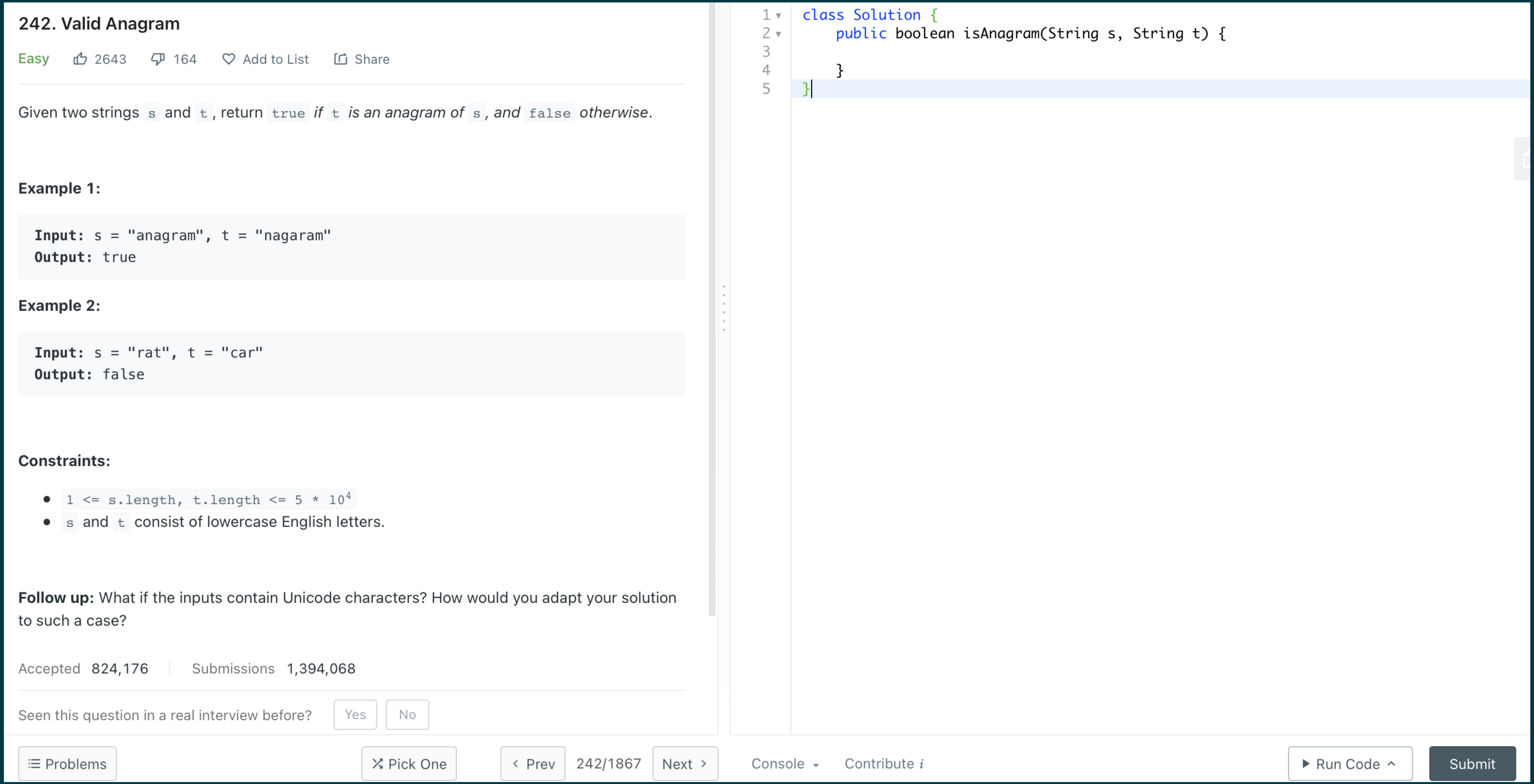
Example 2:

```
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



242. Valid Anagram

Given two strings `s` and `t`, return `true` *if* `t` *is an anagram of* `s`*, and* `false` *otherwise*.

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.

Follow up: What if the inputs contain Unicode characters? How would you adapt your solution to such a case?

Accepted 824,176 | Submissions 1,394,068

Seen this question in a real interview before? Yes No

```java
class Solution {
    public boolean isAnagram(String s, String t) {

    }
}
```

https://leetcode.com/problems/valid-anagram/

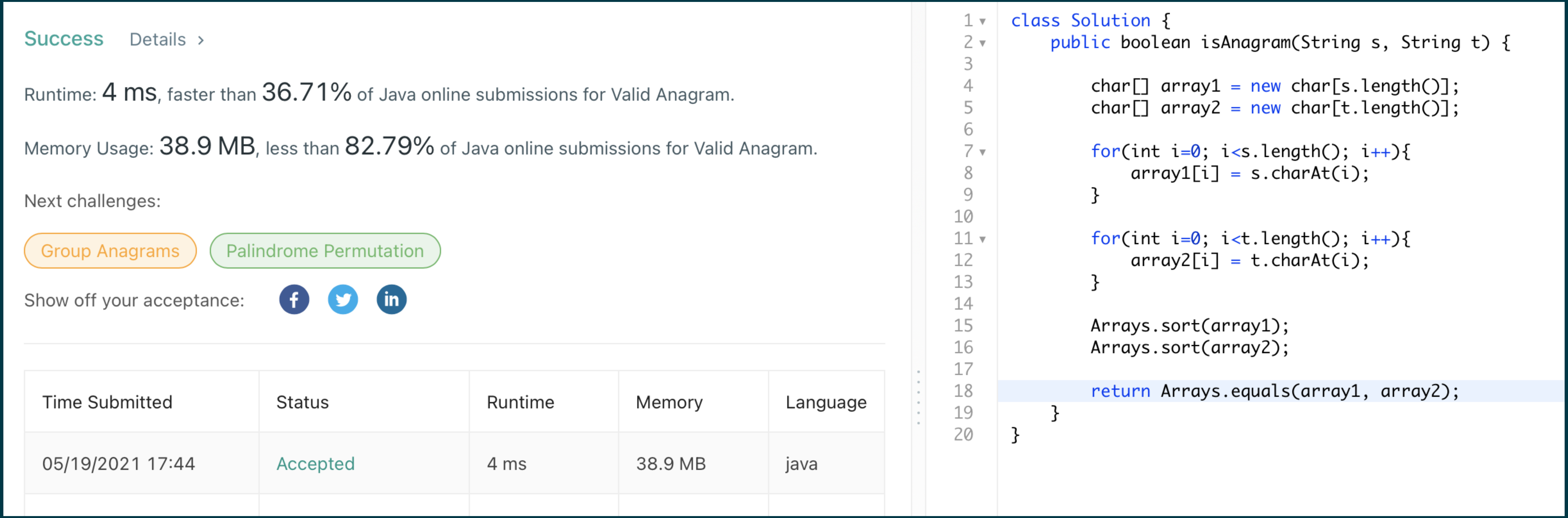
First Theory

S = anagram
T = nagaram

array1 = [a, n, a, g, r, a, m]
array2 = [n, a, g, a, r, a, m]

Sort(array1) = [a, a, a, g, m, n, r]
Sort(array1) = [a, a, a, g, m, n, r]

First Solution



Success Details ›

Runtime: 4 ms, faster than 36.71% of Java online submissions for Valid Anagram.

Memory Usage: 38.9 MB, less than 82.79% of Java online submissions for Valid Anagram.

Next challenges:

Group Anagrams Palindrome Permutation

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
05/19/2021 17:44	Accepted	4 ms	38.9 MB	java

```java
class Solution {
    public boolean isAnagram(String s, String t) {

        char[] array1 = new char[s.length()];
        char[] array2 = new char[t.length()];

        for(int i=0; i<s.length(); i++){
            array1[i] = s.charAt(i);
        }

        for(int i=0; i<t.length(); i++){
            array2[i] = t.charAt(i);
        }

        Arrays.sort(array1);
        Arrays.sort(array2);

        return Arrays.equals(array1, array2);
    }
}
```

Second Theory

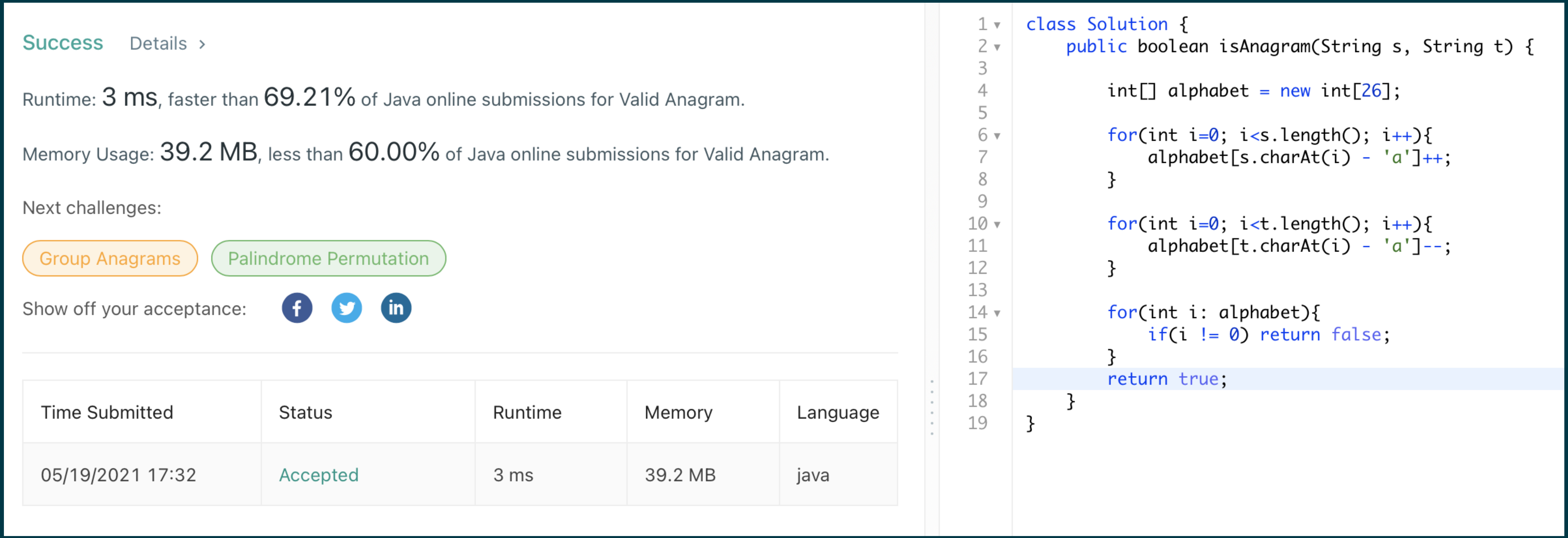
S = anagram
T = nagaram

Lets get empty Box with size 26
Box(26) = [0], [0], [0], … [0], [0], [0], [0]

Put letter of S to Box
Box(26) = [3], [0], [1], … [1], [1], [1], [0]

Remove letter of T from Box
Box(26) = [0], [0], [0], … [0], [0], [0], [0]

Second Solution



Success Details ›

Runtime: 3 ms, faster than 69.21% of Java online submissions for Valid Anagram.

Memory Usage: 39.2 MB, less than 60.00% of Java online submissions for Valid Anagram.

Next challenges:

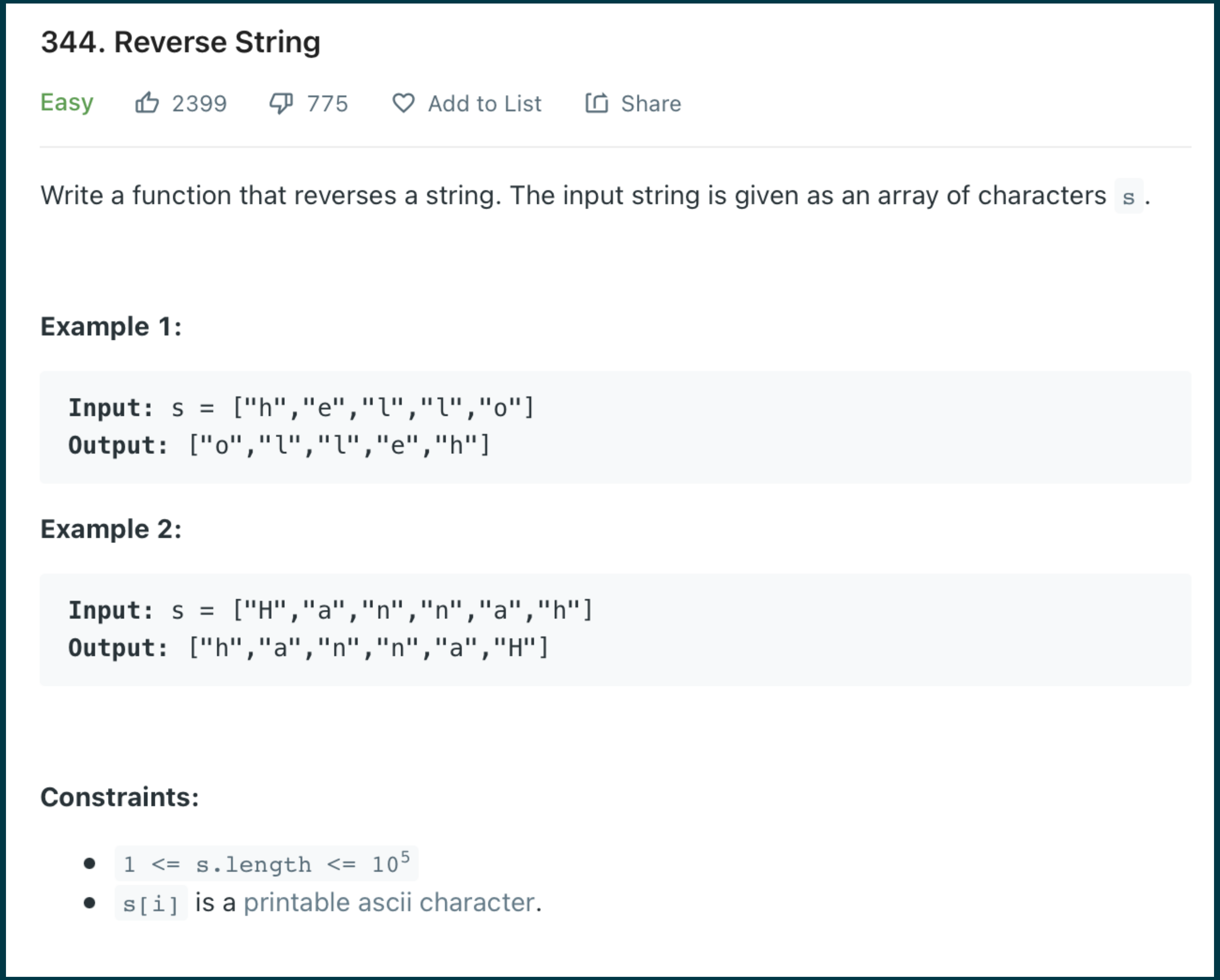
Group Anagrams · Palindrome Permutation

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
05/19/2021 17:32	Accepted	3 ms	39.2 MB	java

```java
class Solution {
    public boolean isAnagram(String s, String t) {

        int[] alphabet = new int[26];

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

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

        for(int i: alphabet){
            if(i != 0) return false;
        }
        return true;
    }
}
```

Task 1 – Reverse String



344. Reverse String

Easy 2399 775 Add to List Share

Write a function that reverses a string. The input string is given as an array of characters `s`.

Example 1:

```
Input: s = ["h","e","l","l","o"]
Output: ["o","l","l","e","h"]
```

Example 2:

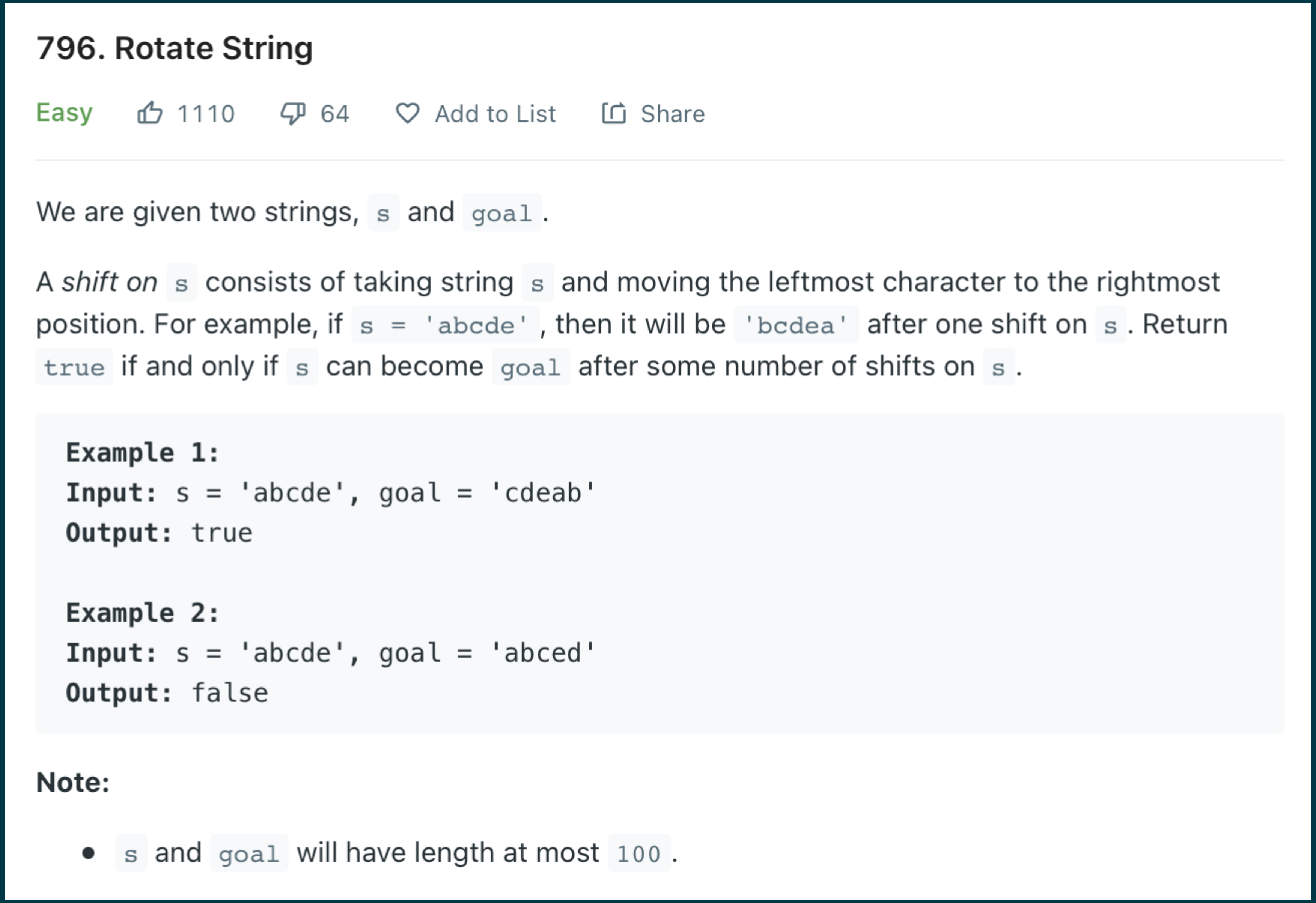
```
Input: s = ["H","a","n","n","a","h"]
Output: ["h","a","n","n","a","H"]
```

Constraints:

- $1 <= s.length <= 10^5$
- `s[i]` is a printable ascii character.

https://leetcode.com/problems/reverse-string/

Task 2 – Rotate String



796. Rotate String

Easy 1110 64 Add to List Share

We are given two strings, `s` and `goal`.

A *shift on* `s` consists of taking string `s` and moving the leftmost character to the rightmost position. For example, if `s = 'abcde'`, then it will be `'bcdea'` after one shift on `s`. Return `true` if and only if `s` can become `goal` after some number of shifts on `s`.

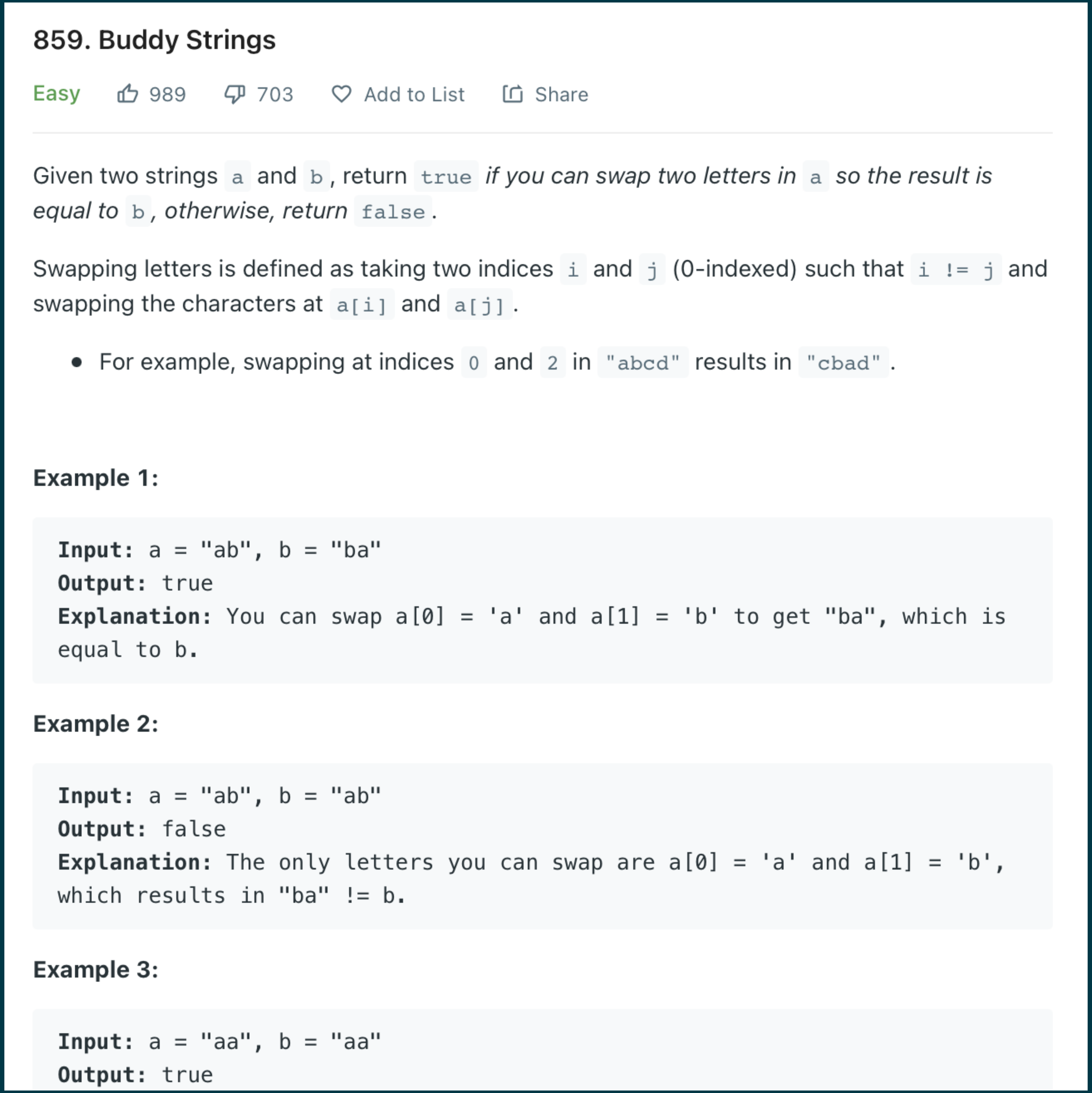
```
Example 1:
Input: s = 'abcde', goal = 'cdeab'
Output: true

Example 2:
Input: s = 'abcde', goal = 'abced'
Output: false
```

Note:

- `s` and `goal` will have length at most `100`.

https://leetcode.com/problems/rotate-string/

Task 3 – Buddy Strings



859. Buddy Strings

Easy 989 703 Add to List Share

Given two strings `a` and `b`, return `true` *if you can swap two letters in* `a` *so the result is equal to* `b`*, otherwise, return* `false`.

Swapping letters is defined as taking two indices `i` and `j` (0-indexed) such that `i != j` and swapping the characters at `a[i]` and `a[j]`.

- For example, swapping at indices `0` and `2` in `"abcd"` results in `"cbad"`.

Example 1:

```
Input: a = "ab", b = "ba"
Output: true
Explanation: You can swap a[0] = 'a' and a[1] = 'b' to get "ba", which is
equal to b.
```

Example 2:

```
Input: a = "ab", b = "ab"
Output: false
Explanation: The only letters you can swap are a[0] = 'a' and a[1] = 'b',
which results in "ba" != b.
```

Example 3:

```
Input: a = "aa", b = "aa"
Output: true
```

https://leetcode.com/problems/buddy-strings/