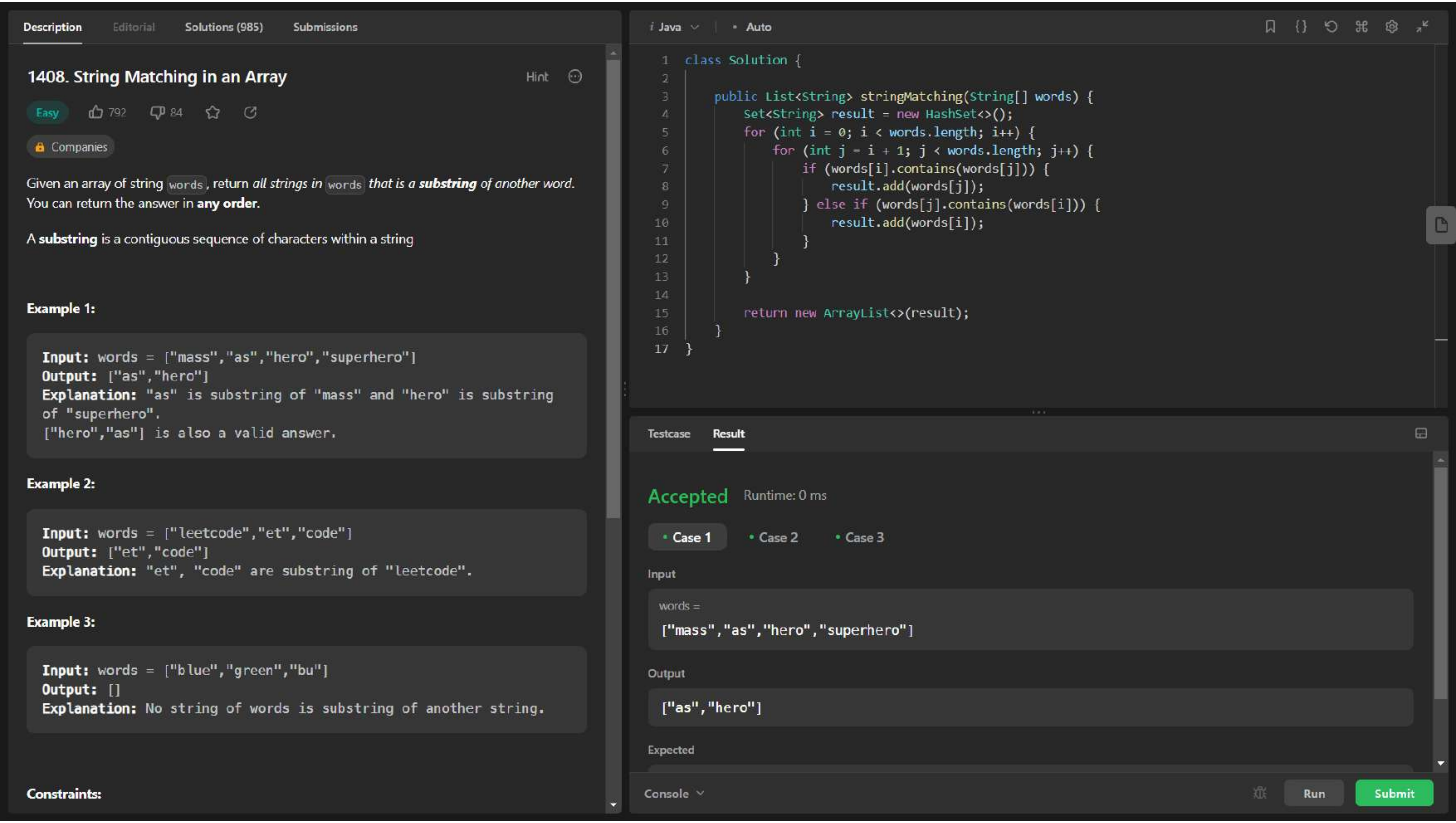


## 1408. String Matching in an Array

Given an array of string `words`, return *all strings in* `words` *that is a* ***substring*** *of another word*. You can return the answer in **any order**.

A **substring** is a contiguous sequence of characters within a string

**Example 1:**

```
Input: words = ["mass","as","hero","superhero"]
Output: ["as","hero"]
Explanation: "as" is substring of "mass" and "hero" is substring
of "superhero".
["hero","as"] is also a valid answer.
```

**Example 2:**

```
Input: words = ["leetcode","et","code"]
Output: ["et","code"]
Explanation: "et", "code" are substring of "leetcode".
```

**Example 3:**

```
Input: words = ["blue","green","bu"]
Output: []
Explanation: No string of words is substring of another string.
```

**Constraints:**

```java
class Solution {

    public List<String> stringMatching(String[] words) {
        Set<String> result = new HashSet<>();
        for (int i = 0; i < words.length; i++) {
            for (int j = i + 1; j < words.length; j++) {
                if (words[i].contains(words[j])) {
                    result.add(words[j]);
                } else if (words[j].contains(words[i])) {
                    result.add(words[i]);
                }
            }
        }

        return new ArrayList<>(result);
    }
}
```

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

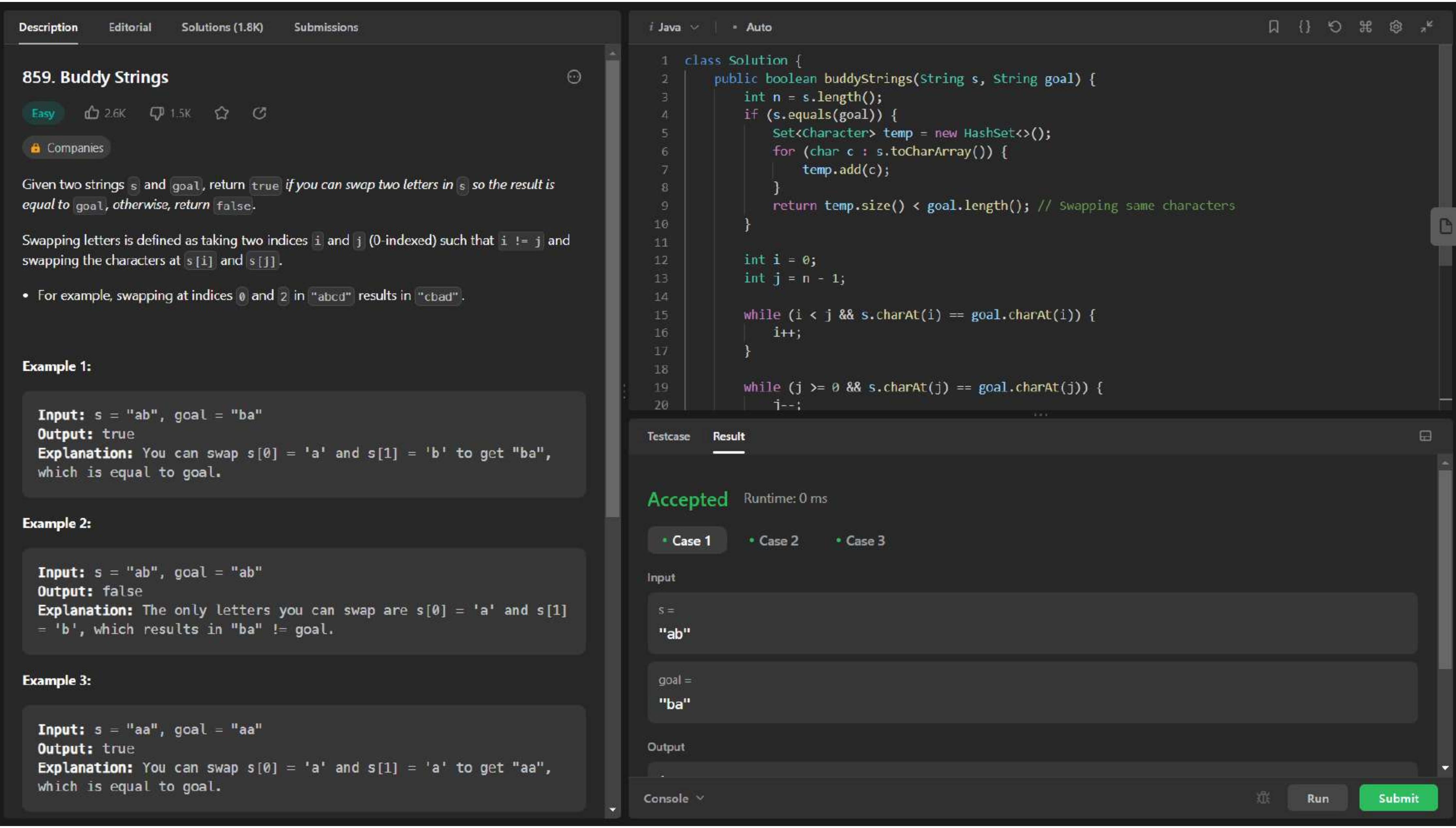
words =

```
["mass","as","hero","superhero"]
```

Output

```
["as","hero"]
```

Expected



## 859. Buddy Strings

Given two strings `s` and `goal`, return `true` *if you can swap two letters in* `s` *so the result is equal to* `goal`*, 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 `s[i]` and `s[j]`.

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

**Example 1:**

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

**Example 2:**

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

**Example 3:**

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

```java
class Solution {
    public boolean buddyStrings(String s, String goal) {
        int n = s.length();
        if (s.equals(goal)) {
            Set<Character> temp = new HashSet<>();
            for (char c : s.toCharArray()) {
                temp.add(c);
            }
            return temp.size() < goal.length(); // Swapping same characters
        }

        int i = 0;
        int j = n - 1;

        while (i < j && s.charAt(i) == goal.charAt(i)) {
            i++;
        }

        while (j >= 0 && s.charAt(j) == goal.charAt(j)) {
            j--;
```

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

s =
"ab"

goal =
"ba"

Output