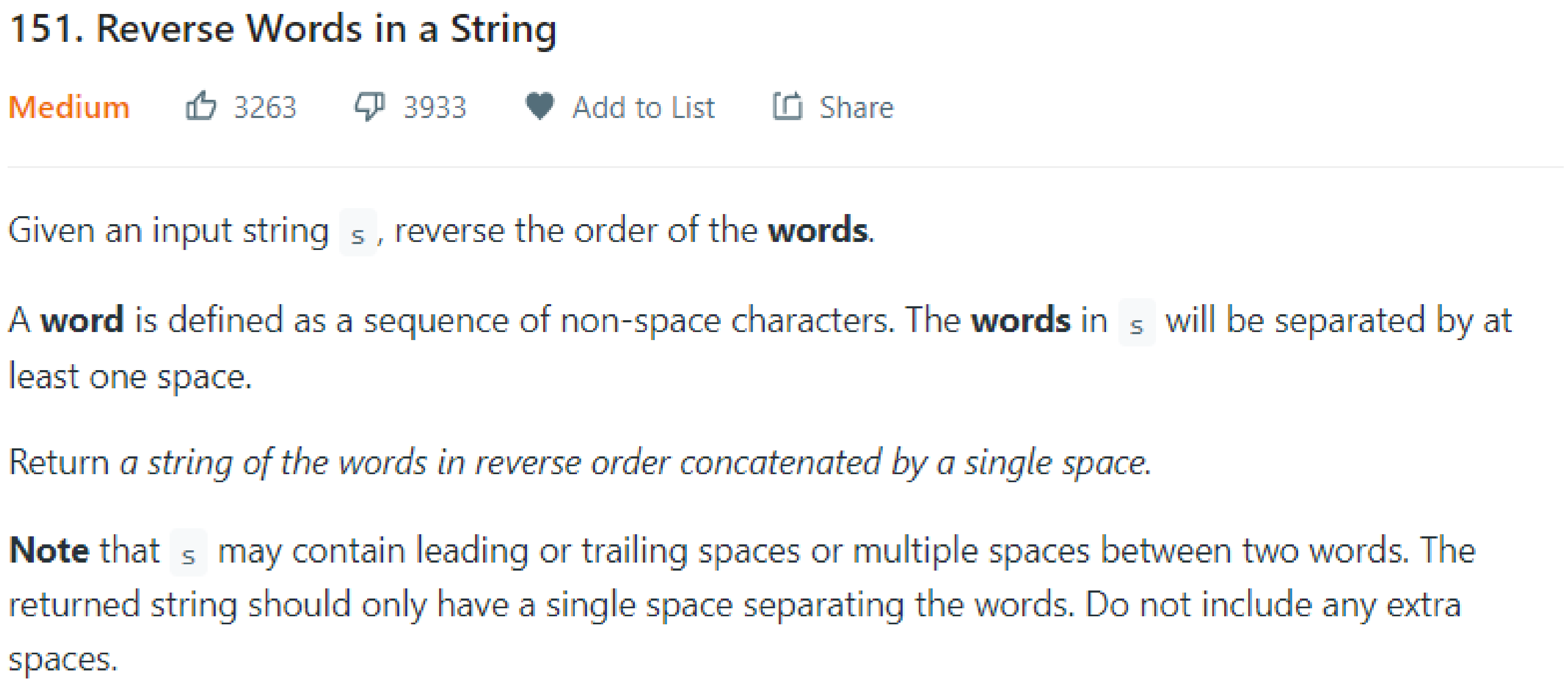


151. Reverse Words in a String

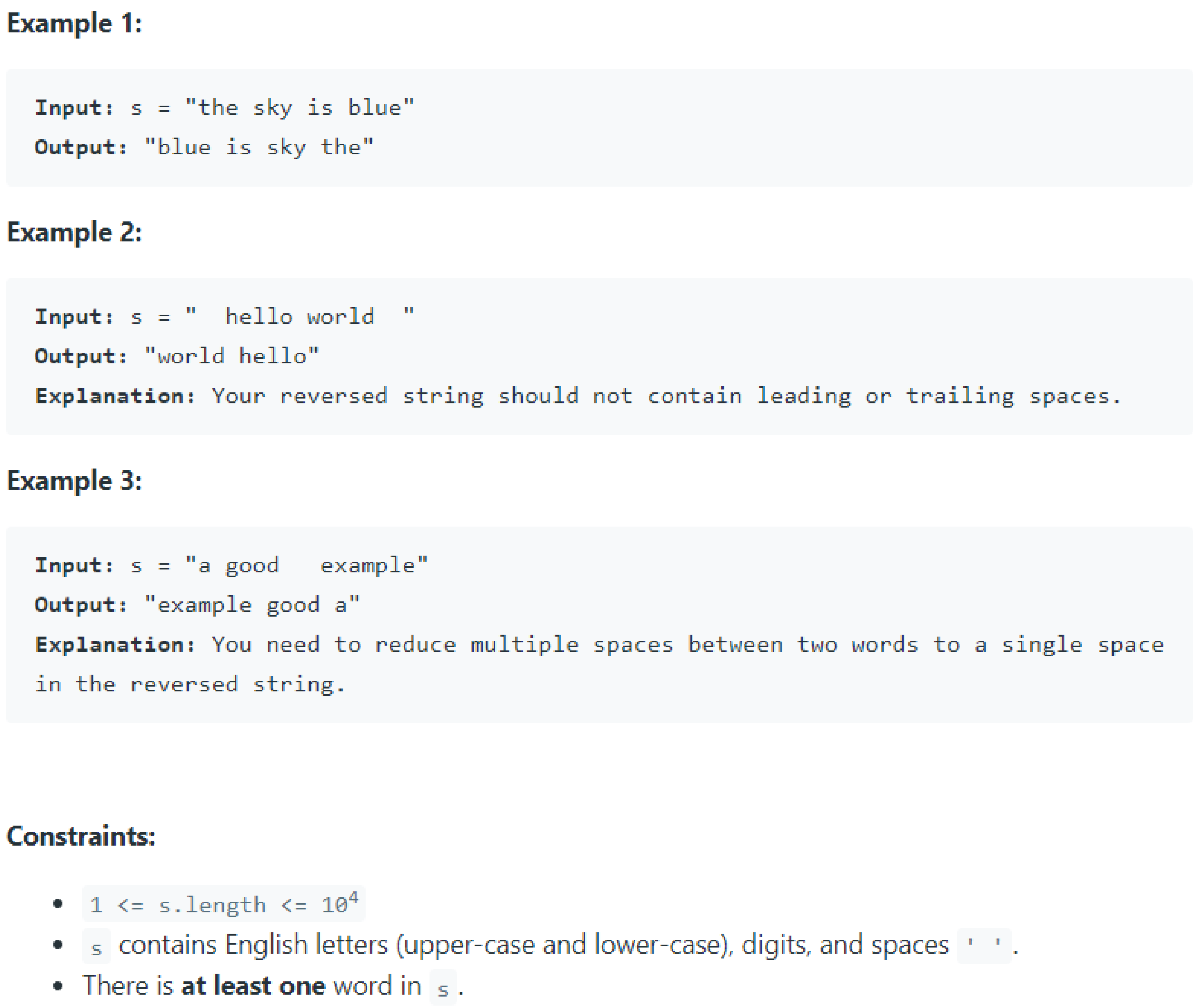
Medium 👍 3263 👎 3933 ♥ Add to List Share

Given an input string `s`, reverse the order of the **words**.

A **word** is defined as a sequence of non-space characters. The **words** in `s` will be separated by at least one space.

Return *a string of the words in reverse order concatenated by a single space.*

Note that `s` may contain leading or trailing spaces or multiple spaces between two words. The returned string should only have a single space separating the words. Do not include any extra spaces.



Example 1:

```
Input: s = "the sky is blue"
Output: "blue is sky the"
```

Example 2:

```
Input: s = "  hello world  "
Output: "world hello"
Explanation: Your reversed string should not contain leading or trailing spaces.
```

Example 3:

```
Input: s = "a good   example"
Output: "example good a"
Explanation: You need to reduce multiple spaces between two words to a single space
in the reversed string.
```

Constraints:

- `1 <= s.length <=` 10^4
- `s` contains English letters (upper-case and lower-case), digits, and spaces `' '`.
- There is **at least one** word in `s`.

```
string reverseWords(string s) {
    if(s.size() == 0) {
        return s;
    }

    stack<string> stack;
    string result;

    for(int i=0; i<s.size(); i++) {
        string word;
        if(s[i] == ' ') continue;

        while((i < s.size()) && (s[i] != ' ')) {
            word += s[i];
            i++;
        }
        stack.push(word);
    }

    while(!stack.empty()) {
        result += stack.top();
        stack.pop();
        if(!stack.empty()) {
            result += " ";
        }
    }
    return result;
}
```

#100daysofDSA

/rvislive

Rakesh Vishwakarma