**Reverse Words in a String**

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.

class Solution {

    public String reverseWords(String s) {

        if (s == null || s.length() == 0) {

            return "";

        }

        String[] words = s.split(" ");

        StringBuilder sb = new StringBuilder();

        for (int i = words.length - 1; i >= 0; i--) {

            if (!words[i].equals("")) {

                sb.append(words[i]);

                sb.append(" ");

            }

        }

        if (sb.length() == 0) {

            return "";

        }

        return sb.substring(0, sb.length() - 1);

    }

}