[094]152.Reverse word in a string I

```
- Date:2020-12-22(Tue)
- medium
- Related topic:String
```

problem link:https://leetcode.com/problems/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.

Example 4:

```
Input: s = "  Bob    Loves  Alice   "
Output: "Alice Loves Bob"
```

Example 5:

```
Input: s = "Alice does not even like bob"
Output: "bob like even not does Alice"
```

Constraints:

- 1 <= s.length <= 104
- s contains English letters (upper-case and lower-case), digits, and spaces ' '.
- There is at least one word in s.

Think Process:

- 首先把input S作處理，並且存進去String[]內， "Hi Leetcode"變成，["Hi", "Leetcode"]
- 然後開始逆向traverse當中的array，每一組都要加上“ ”(空格)
- 最後，在做trim的處理

Code:

```java
class Solution {
    public String reverseWords(String s) {
        StringBuilder sb = new StringBuilder();
        String[] words = s.trim().split("\\s+");
        for(int i=words.length-1;i>=0;i--){
            sb.append(words[i]+ " ");
        }
        String res = sb.toString();
        return res.trim();
    }
}
```

Complexity Analysis

- Time complexity: O(N), where N is a number of characters in the input string.
- Space complexity: O(N), to store the result of split by spaces.

Result:

- Runtime: 8 ms, faster than 45.14% of Java online submissions for Reverse Words in a String.
- Memory Usage: 39.2 MB, less than 55.42% of Java online submissions for Reverse Words in a String.

Next challenges:

- Substring with Concatenation of All Words
- Output Contest Matches
- Count the Number of Consistent Strings