# **Leetcode 151 – Reverse Words in a String**

## Problem Understanding

Given a string s, **reverse the order of words** in the string.

* A **word** is defined as a sequence of non-space characters.
* You must **remove leading/trailing spaces** and **reduce multiple spaces between words to a single space** in the result.

### Examples:

Input: s = " hello world "

Output: "world hello"

## Optimized Java Solution

class Solution {

public String reverseWords(String s) {

s = s.trim(); // Remove leading/trailing spaces

String[] words = s.split("\\s+"); // Split by 1+ spaces

StringBuilder sb = new StringBuilder();

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

sb.append(words[i]);

if (i != 0) sb.append(" ");

}

return sb.toString();

}

}

## Dry Run Using Table

### Input: " the sky is blue "

|  |  |  |
| --- | --- | --- |
| Step | Action | Result / Comment |
| Trim | "the sky is blue" | Leading/trailing removed |
| Split | ["the", "sky", "is", "blue"] | Multiple spaces removed |
| Reverse | ["blue", "is", "sky", "the"] | Order reversed |
| Join | "blue is sky the" | Final output |

✅ Final Answer: "blue is sky the"

## Time / Space Complexity

|  |  |
| --- | --- |
| Metric | Value |
| Time | O(n) |
| Space | O(n) |

* Splitting, trimming, and reversing all take linear time and space.

## Alternate Approaches

|  |  |  |  |
| --- | --- | --- | --- |
| Approach | Time | Space | Notes |
| ✅ Built-in Split & Join | O(n) | O(n) | Clean, fast, recommended |
| Manual Reverse + Char Array | O(n) | O(n) | Slightly harder, used in interviews |
| Stack-based approach | O(n) | O(n) | Can also be done using a stack |