Reverse Words in a String [LeetCode](https://leetcode.com/problems/reverse-words-in-a-string/description/)

Given a string containing multiple words separated by spaces, the task is to reverse the order of the words in the string.

Example: Input: " Hello World "

Output: “World Hello”

**Approach 1: Reverses the words in the input string using a vector**

* The function iterates over the input string character by character and builds each word.
* The words are stored in a vector in the order they appear in the input string.
* Finally, the function constructs the reversed string by appending the words in reverse order.
* **Time Complexity: O(n), where n is the length of the input string. It involves iterating over the string once and appending words to the vector.**
* **Space Complexity: O(m), where m is the number of words in the input string. The function uses additional space to store the words in the vector.**

**Approach 2: Reverses the words in the input string using a stack**

* The function follows a similar approach to the vector approach.
* Instead of storing words in a vector, it uses a stack to reverse the order of the words.
* Words are pushed onto the stack while iterating over the input string and then popped from the stack to construct the reversed string.
* **Time Complexity: O(n), where n is the length of the input string. It involves iterating over the string once and pushing/popping words onto/from the stack.**
* **Space Complexity: O(m), where m is the number of words in the input string. The function uses additional space to store the words in the stack.**

**Approach 3: Reverses the words in the input string using another string**

* The function uses two pointers to traverse the input string from right to left.
* It extracts each word from the string while skipping trailing spaces.
* The reversed string is constructed by appending each word in reverse order with spaces in between.
* **Time Complexity: O(n), where n is the length of the input string. It involves iterating over the string once and constructing the reversed string.**
* **Space Complexity: O(n), where n is the length of the input string. The function uses additional space to store the reversed string.**