# 151. 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.

## SOLUTION IN C++

class Solution {

public:

string reverseWords(string s) {

reverse(s.begin(), s.end()); // Reverse the whole string.

reverseWords(s, s.length()); // Reverse each word.

return cleanSpaces(s, s.length()); // Clean up the spaces.

}

private:

void reverseWords(string& s, int n) {

int i = 0;

int j = 0;

while (i < n) {

while (i < j || i < n && s[i] == ' ') // Skip the spaces.

++i;

while (j < i || j < n && s[j] != ' ') // Skip the spaces.

++j;

reverse(s.begin() + i, s.begin() + j); // Reverse the word.

}

}

// Trim leading, trailing, and middle spaces

string cleanSpaces(string& s, int n) {

int i = 0;

int j = 0;

while (j < n) {

while (j < n && s[j] == ' ') // Skip the spaces.

++j;

while (j < n && s[j] != ' ') // Keep non spaces

s[i++] = s[j++];

while (j < n && s[j] == ' ') // Skip the spaces.

++j;

if (j < n) // Keep only one space.

s[i++] = ' ';

}

return s.substr(0, i);

}

};