151. Reverse Words in a String

Created	@June 10, 2021 1:14 PM
Tags	Medium
link	https://leetcode.com/problems/reverse-words-in-a-string/

Description:

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.

solve it in-place with O(1) extra space

Approach:

On the surface the problem looks very easy. The difficulty arises from the fact of doing in O(1) extra space.

- Notice that if we reverse whole `input string` and then reverse each word in it, we arrive at result.
- So, steps become
    - Remove duplicate spaces
    - Reverse whole spaces
    - Identify individual words and reverse them

```
class Solution {
public:
    bool isValid(char c) {
        if (c - 'a' >= 0 && c - 'a' <= 25) return true; // lower case
        if (c - 'A' >= 0 && c - 'A' <= 25) return true; // upper case
        if (c - '0' >= 0 && c - '0' <= 10) return true; // digit

        return false;
    }

    string removeDuplicateSpaces(string s) {
        string result = "";
        int i = 0;
        while (i < s.size() && s[i] == ' ') i++; // remove trailing spaces

        while (i < s.size()) {
            int j = i;
            while (j < s.size() && isValid(s[j])) result += s[j++];

            result += ' ';

            while (j < s.size() && s[j] == ' ') j++; // remove duplicate spaces

            i = j;
        }

        return result.substr(0, result.size() - 1);
    }

    string reverseWords(string s) {
        s = removeDuplicateSpaces(s);
        reverse(s.begin(), s.end());
        // Identify individual words and reverse
        for (int i = 0; i < s.size();) {
            int j = i;
            while (j < s.size() && isValid(s[j])) j++;
            reverse(s.begin() + i, s.begin() + j);
            i = j+1;
        }

        return s;
    }
};
```