**Strings**

**Reverse String**

Write a function that reverses a string. The input string is given as an array of characters s.

You must do this by modifying the input array [in-place](https://en.wikipedia.org/wiki/In-place_algorithm) with O(1) extra memory.

Example 1:

Input: s = ["h","e","l","l","o"]

Output: ["o","l","l","e","h"]

Key to problem:

Use the 2-pointer approach –> O(n) time since only run through list once; O(1) space since swap in-place

Move left and right pointers through until both pointers point to the same position

A picture containing text, screenshot, font, line

Description automatically generated

Other methods: <https://www.youtube.com/watch?v=_d0T_2Lk2qA>

1. Stack – Add elements to stack one by one which reverses string, then pop each element from stack to add back to array; O(n) time, but O(n) space

|  |
| --- |
| O |
| L |
| L |
| E |
| H |

[‘H’, ‘E’, ‘L’, ‘L’, ‘O’]

[‘O’, ‘L’, ‘L’, ‘E’, ‘H’]

A screenshot of a computer program

Description automatically generated with low confidence

1. Recursion – recursive portion are the chars other than the 1st and last elements

O(n) time, but O(n) space

A screenshot of a computer code

Description automatically generated with low confidence

**Reverse Integer**

Given a signed 32-bit integer x, return xwith its digits reversed. If reversing x causes the value to go outside the signed 32-bit integer range [-231, 231 - 1], then return 0.

**Assume the environment does not allow you to store 64-bit integers (signed or unsigned).**

**Example 1:**

**Input: x = 123**

**Output: 321**

<https://www.youtube.com/watch?v=HAgLH58IgJQ\>

A screenshot of a computer program

Description automatically generated with medium confidence

**First Unique Character in a String**

Given a string s, find the first non-repeating character in it and return its index. If it does not exist, return -1.

Functions used to solve:

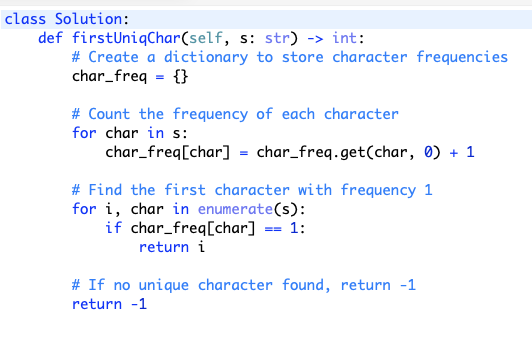
The char\_freq.get(char, 0) method is used to retrieve the value associated with the char key in the char\_freq dictionary. If the key char is present in the dictionary, the method returns its corresponding value. Otherwise, it returns the default value provided as the second argument, which is 0.

The enumerate() function in Python is used to iterate over a sequence while keeping track of the index of each element. It returns pairs of (index, element) as you iterate through the sequence.

Complexity:

Time: O(n) - iterate over the string once to count the frequency of each character, and then iterate over it again to find the first character with a frequency of 1. Each op has linear time complexity.

Space: O(k) where k is the number of unique characters in the input string s



**Valid Anagram**

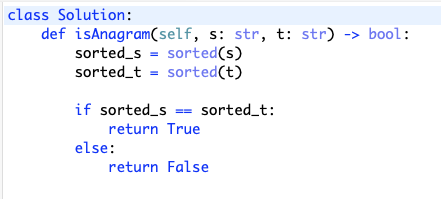
Given two strings s and t, return true if t is an anagram of s, and false otherwise.

An **Anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

Method 1: Use sorted()

Time: O(nlogn)

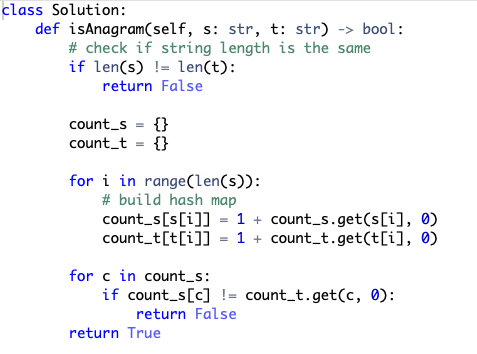
Space: O(n)



Method 2: Count each char using hash map; key = (char, count)

Compare counts of each letter

Time and space: O(S + T)



**Valid Palindrome**

A phrase is a **palindrome** if, after converting all uppercase letters into lowercase letters and removing all non-alphanumeric characters, it reads the same forward and backward. Alphanumeric characters include letters and numbers.

Given a string s, return trueif it is a***palindrome***, orfalseotherwise.

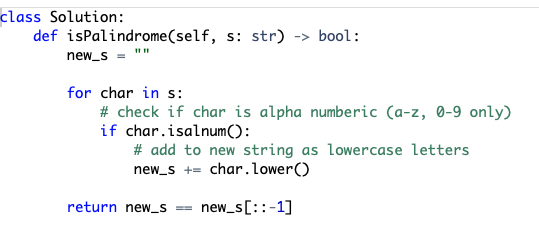
Input: s = "A man, a plan, a canal: Panama"

Output: true

Explanation: "amanaplanacanalpanama" is a palindrome.

<https://www.youtube.com/watch?v=jJXJ16kPFWg>

Method 1: Use built-in methods to check whether char is alpha numeric and to change all chars to lower case



Method 2: Double pointers – wouldn’t use extra memory like above

Use a left and right pointer which you’ll increment/ decrement until they meet in the middle

Time: O(n)

Space: O(1)

A screenshot of a computer program

Description automatically generated with medium confidence

**String to Integer (atoi)**

Implement the myAtoi(string s) function, which converts a string to a 32-bit signed integer (similar to C/C++'s atoi function).

The algorithm for myAtoi(string s) is as follows:

1. Read in and ignore any leading whitespace.
2. Check if the next character (if not already at the end of the string) is '-' or '+'. Read this character in if it is either. This determines if the final result is negative or positive respectively. Assume the result is positive if neither is present.
3. Read in next the characters until the next non-digit character or the end of the input is reached. The rest of the string is ignored.
4. Convert these digits into an integer (i.e. "123" -> 123, "0032" -> 32). If no digits were read, then the integer is 0. Change the sign as necessary (from step 2).
5. If the integer is out of the 32-bit signed integer range [-231, 231 - 1], then clamp the integer so that it remains in the range. Specifically, integers less than -231 should be clamped to -231, and integers greater than 231 - 1 should be clamped to 231- 1.
6. Return the integer as the final result.

**Note:**

* Only the space character ' ' is considered a whitespace character.
* **Do not ignore** any characters other than the leading whitespace or the rest of the string after the digits.

**Input:** s = "42"

**Output:** 42

**Explanation:** The underlined characters are what is read in, the caret is the current reader position.

Step 1: "42" (no characters read because there is no leading whitespace)

^

Step 2: "42" (no characters read because there is neither a '-' nor '+')

^

Step 3: "42" ("42" is read in)

^

The parsed integer is 42.

Since 42 is in the range [-231, 231 - 1], the final result is 42.

Key to problem:

Identify all edge cases before jumping in

Time: O(n)

Space: O(1)

<https://www.youtube.com/watch?v=YA0LYrKI1CQ>

A screenshot of a computer code

Description automatically generated with low confidence

**Implement strStr()**

Given two strings needle and haystack, return the index of the first occurrence of needle in haystack, or -1 if needle is not part of haystack.

Input: haystack = "sadbutsad", needle = "sad"

Output: 0

Explanation: "sad" occurs at index 0 and 6.

The first occurrence is at index 0, so we return 0.

Key to problem:

Nested for loops – brute force solution

Time: O(n\*m) where n and m are sizes of both strings

Space: O(1)

A screenshot of a computer program

Description automatically generated with medium confidence

A picture containing text, font, screenshot, algebra

Description automatically generated

<https://www.youtube.com/watch?v=Gjkhm1gYIMw&t=11s>

**Longest Common Prefix**

Write a function to find the longest common prefix string amongst an array of strings.

If there is no common prefix, return an empty string "".

Example 1:

Input: strs = ["flower","flow","flight"]

Output: "fl"

Time: O(n \* m)

A screenshot of a computer code

Description automatically generated with low confidence