

1. String to Integer (atoi).

Example 1: 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 $[-2^{31}, 2^{31} - 1]$, the final result is 42.

Example 2: Input: s = " -42" Output: -42

Explanation: Step 1: " -42" (leading whitespace is read and ignored)

Step 2: " -42" ('-' is read, so the result should be negative) Step 3: " -42" ("42" is read in)

The parsed integer is -42. Since -42 is in the range $[-2^{31}, 2^{31} - 1]$, the final result is -42.

2. 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"

Example 2:

Input: strs = ["dog", "racecar", "car"]

Output: ""

Explanation: There is no common prefix among the input strings.

3. Divide Two Integers

Given two integers dividend and divisor, divide two integers **without** using multiplication, division, and mod operator.

4. Count and Say.

The count-and-say sequence is a sequence of digit strings defined by the recursive formula: countAndSay(1) = "1" countAndSay(n) is the way you would "say" the digit string from countAndSay(n-1), which is then converted into a different digit string.

To determine how you "say" a digit string, split it into the minimal number of substrings such that each substring contains exactly one unique digit. Then for each substring, say the number of digits, then say the digit. Finally, concatenate every said digit. For example, the saying and conversion for digit string "3322251":

Example 1: Input: n = 1 Output: "1" Explanation: This is the base case.

Example 2: Input: n = 4 Output: "1211"

Explanation :countAndSay(1) = "1" countAndSay(2) = say "1" = one 1 = "11" countAndSay(3) = say "11" = two 1's = "21" countAndSay(4) = say "21" = one 2 + one 1 = "12" + "11" = "1211"

5. Maximum Subarray.

Given an integer array nums, find the subarray with the largest sum, and return its sum.

Example 1: Input: nums = [-2,1,-3,4,-1,2,1,-5,4] Output: 6

Explanation: The subarray [4,-1,2,1] has the largest sum 6.

Example 2: Input: nums = [1] Output: 1

Explanation: The subarray [1] has the largest sum 1.

Example 3: Input: nums = [5,4,-1,7,8] Output: 23

Explanation: The subarray [5,4,-1,7,8] has the largest sum 23.

6. Length of small Word except space character.

i/p: luffy is still joyboy.

o/p:is

7. Greatest Sum Divisible by Three.

Given an integer array nums, return the maximum possible sum of elements of the array such that it is divisible by three.

Example 1: Input: nums = [3,6,5,1,8] Output: 18

Explanation: Pick numbers 3, 6, 1 and 8 their sum is 18 (maximum sum divisible by 3).

Example 2: Input: nums = [4] Output: 0

Explanation: Since 4 is not divisible by 3, do not pick any number.

8. Element appearing more than the 25% of the time ,return that integer.

Given an integer array sorted in non-decreasing order, there is exactly one integer in the array that occurs more than 25% of the time, return that integer.

Example 1:Input: arr = [1,2,2,6,6,6,6,7,10] Output: 6

Example 2: Input: arr = [1,1] Output: 1