(1) For the length of the last word → Easy one

Split the string into words using spaces as the delimeter.

words = s.split()

If there are no words, return 0,

then Return the length of the last word in the list of words.

Return len(words[-1]).

(1) Define the function ie, length_of_last_word

(2) Split the strings into words

(3) check if the list of words ('words') is empty. if it is zero. there are no words

(4) Return the length of last word.

# Majority Element → Medium ②

Algo Algorithm :-

① Initialize candidate1, candidate2 equal to None
And count1, count2 equal to 0.

② The input list nums'
if number is equal to Candidate1, increment count1
if number is equal to candidate2, increment count2
if count1 is 0, update candidate1, set count1 = 1
if count2 is 0, update candidate2, set count2 to 1

③ Reset count1, count2 to 0

④ check candidate1, candidate2 are > [n/3]
If Candidate1 > n/3, add it to the result
if Candidate2 > n/3 add it to result

⑤ Result list is returns.

· Find the total number of digit 1 → Hard 3

Algorithm

① Count and add 1's in - one's place, Tenth place, Hundredth place

one's place

1 11 21 31 41 51 61 71 81 91

upto 10 → 1 one
upto 20 → 2 one
upto 80 → 8 one
upto 81 → 9 one

Number of 1's in ones place = $(n/10) + n\%10 != 0)$

Ten's place ⇒ upto 100 → 10 one
200 → 20 one
⋮
upto 800 → 80 one

Number of 1's in tenth place = $(n/100)*10 + \min(\max(n\%100 - 10 + 1, 0), 10)$

② Iterate from 1 to n by incrementing 10.

This is formulated as

$(n/(i*10))*i + \min(\max((n\%(i*10)) - i + 1, 0), i)$

adding each iteration).