Deep Arithmetic Published by **nihiltres** in Java 🔻 arrays language_fundamentals objects recursion Create a function that takes an array of strings of arbitrary dimensionality (String[], String[][], String[][]], etc) and returns the sum of every separate number in each string in the array. Examples sum({"1", "five", "2wenty", "thr33"}) → 36 ""(1 {"1X2", "t3n"}, {"1024", "5", "64"} $1) \rightarrow 1099$ sum({ {"0", "0x2", "z3r1"}, {"1", "55a46"} {"1", "2", "4"}, {"0x5fp-2", "nine", "9"}, {"4", "4", "4"} {"03"} }) → 142

We can ascertain the number rows in jagged array by performing sum.length We can ascertain the number of elements in each row by performing sum[row number].length

It is a case of performing this recursively...

We will let r equal row number and let n equal to the element in row r.

it will require .charAt on each sum[r][n] element.

it will perform a regex check for a digit 0-9 on the last character on the substring.

if it finds a digit, it will perform a check again (recursively) to identify if consecutive digits.

if it doesn't find a consecutive digit, at this point it is ready to establish this as the full number. And perform addition to running total.

This appears to be a nested recursive loop

I foresee my first major challenge here, since we are potentially using recursion to traverse between elements sum[r][n]..

However we need to bear in mind that recursion will also occur whilst using substring on the existing sum[r][n] when performing truncation and identifying the digits..

Since we are dealing with a function as per the challenge requirements, recurrence will occur on the basis of three conditions. Is it possible to drive this logic into one recursive call with same signature? At what point do we utilize iteration.

In several of my challenges, I have not been able to conform to a single method / function with recursive challenges since the coding becomes too deep and unless several booleans are introduced, it is not possible to get this deep.

I had a similar issue with addition problem using pure recursion. I had to switch to a hybrid approach. Perhaps I need to think about this a bit more before using a hybrid approach.

Also I had doubt on what exactly is iteration.. Is it using an incrementor or decrement or to the advantage of accessing indexes... Or would it not be permissible if I used an incrementor supposedly in a for each loop to control the flow and access certain element?