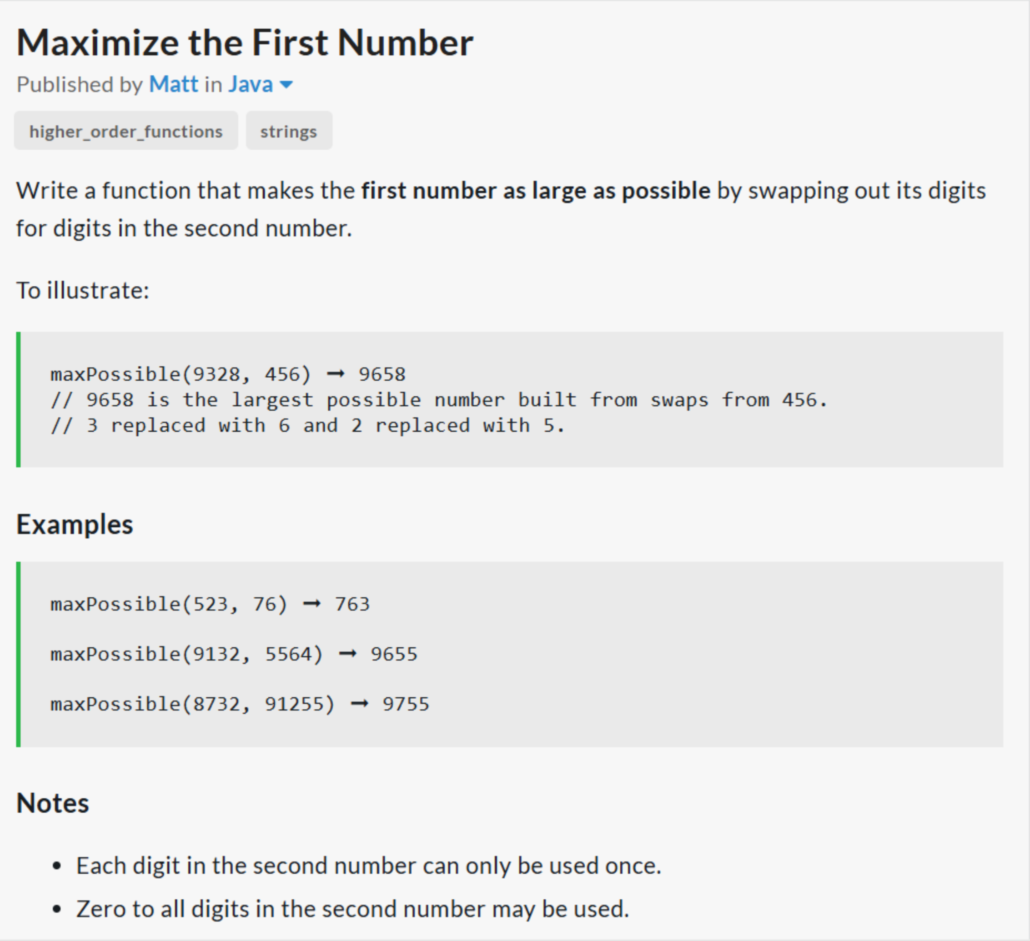

Maximize the First Number

Published by Matt in Java

higher_order_functions strings

Write a function that makes the **first number as large as possible** by swapping out its digits for digits in the second number.

To illustrate:

```
maxPossible(9328, 456) ➞ 9658
// 9658 is the largest possible number built from swaps from 456.
// 3 replaced with 6 and 2 replaced with 5.
```

Examples

```
maxPossible(523, 76) ➞ 763

maxPossible(9132, 5564) ➞ 9655

maxPossible(8732, 91255) ➞ 9755
```

Notes

- Each digit in the second number can only be used once.
- Zero to all digits in the second number may be used.

This would simply involve recursion in which the length of the initial number is traversed. We know maximum value is an offset of exchanging MSD first with largest number available.
Utilising charAt on the second number and changing to integer.
And performing a character removal.
There will be two options, work on the String from offset and transfer content into another String.
Easier would be to create a StringBuilder since we know String is immutable. Add the String into the StringBuilder.
And can perform sb.remove(....)

I am still not convinced it will be this simple.
And also unaware of any forthcoming challenges prior to the implementation.