

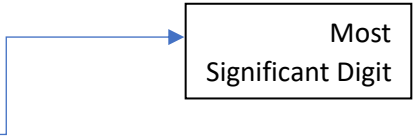
Largest number formed from an array of positive integers

Let $A = \{a_1, a_2, \dots, a_n\}$ be an array of positive integers.

Assume A is sorted in a non-increasing order

Let r be the required result.

Most
Significant Digit



Assuming we're pointing to the i' th MSD of r , and the other MSD have been already populated.

The question we would like to answer is:

What $a \in A$ we need to choose such that the i' th MSD of r will be the largest.

Let $L = \{l_1, l_2, \dots, l_m\}$ s.t $1 \leq m \leq n$ be an array of positive integers. We say for some l that $l \in L$ if $\exists a \in A$ s.t the number of digits in a is l .

For example:

(*) $A = \{3, 34, 54, 5433\}$

$L = \{1, 2, 4\}$

We can initialize an array of ordered pairs of non-negative integers OP .

We say for some i, j s.t $1 \leq i \leq j$ that $(i, j) \in OP$ if there're $j - i + 1$ integers in A from index i to j s.t these integers contains the same number of digits.

For example, in (*), $OP = \{(1,1), (2,3), (4,4)\}$.

Assuming that the **first** integer that we choose from A for r is $currMax = 0$.

and $currMaxLength = 1$

For each $(i, j) \in OP$ we'll start from $A[j]$ and compare it with $currMax$.

We compare those integers according to their MSD:

if(current MSD of $A[j]$ < current MSD of $currMax$)

(**) *then do:*

$currMax \leftarrow A[j]$

$currMaxLength = \text{length of } A[j]$

return true / acknowledge that a change were made*/*

else if(current MSD of $A[j]$ = current MSD of $currMax$)

then compare current MSD - 1 of $A[j]$ to

current MDS - 1 of $currMax$

else

*return false /*no change were made*/*

if current MSD of currMax = 0 then

if current MSD of $A[j]$ = 0 then return false / they're equal, no change were made */*

else do current MSD of currMax = currMaxLength

*else if current MSD of $A[j]$ = 0 then do current MSD of $A[j]$ =
length of $A[j]$*

All of the conditions above will be put in a loop.

There's some interesting case to mention.

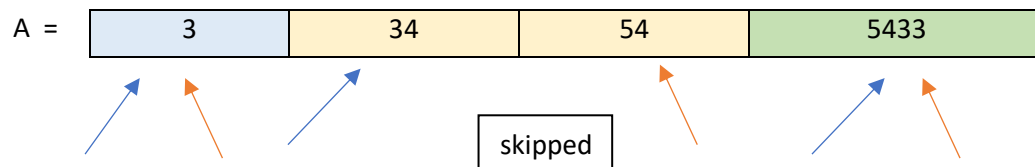
if there're integers $a = a_n a_{n-1} \dots a_2 a_1$ and $b = b_k b_{k-1} \dots b_i b_{i-1} \dots b_2 b_1$ s.t

$a_n a_{n-1} \dots a_2 a_1 = b_k b_{k-1} \dots b_i$ then we need to compare a to $b_{i-1} \dots b_2 b_1$

so, we'll exit the loop when we finish to compare all the remaining digits.

if we finish to iterate all the digits of some integer a but not the other integer, then we'll continue to compare the digits of the two integers but we'll reset the digit counter of a . Thus we'll point to the MSD of a again.

Example:



until you didn't populate all digits of r do :

if the current orange arrow didn't skip the current blue arrow do :

currMax = 0, currMaxLength = 1;

compare first digit of 3 to first digit of 0 → 3 wins

currMax = 3, currMaxLength = 1;

compare first digit of 3 to first digit of 54 → 5 wins

currMax = 54, currMaxLength = 2;

compare first digit of 54 to first digit of 5433 → even

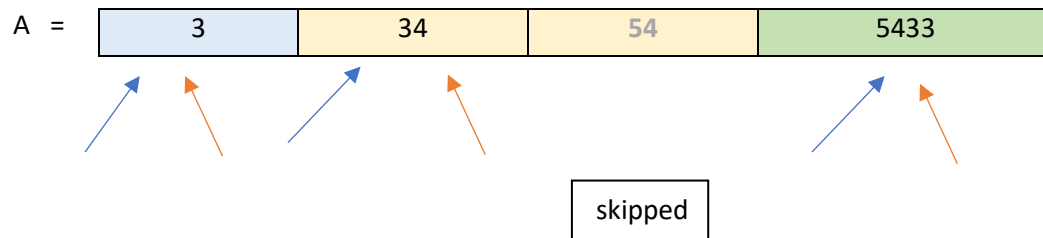
compare second digit of 54 to second digit of 5433 → even

do MSD of 54 = 2

compare first digit of 54 to third digit of 5433 → 54 wins

first integer for r is 54

$r = 54$



currMax = 0, currMaxLength = 1;

compare first digit of 3 to first digit of 0 → 3 wins

currMax=3, currMaxLength=1

compare first digit of 3 to first digit of 34 → even

do MSD of 3 = 1

compare first digit of 3 to second digit of 34 → 34 wins

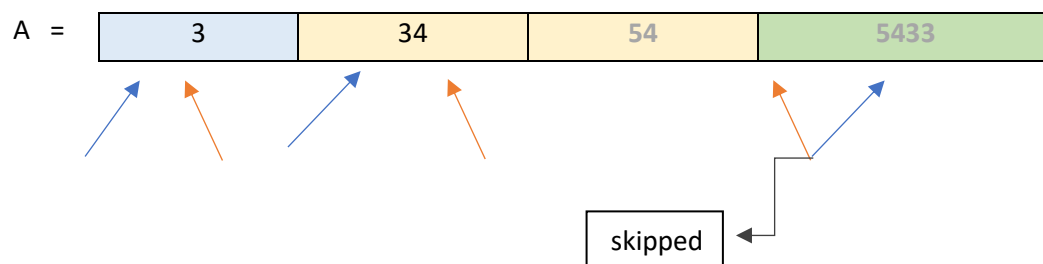
currMax = 34, currMaxLength = 2;

compare first digit of 34 to first digit of 5433 → 5433 wins

currMax = 5433, currMaxLength = 4

second integer for r is 5433

$r = 545433$



currMax = 0, currMaxLength = 1;

compare first digit of 3 to first digit of 0 → 3 wins

currMax=3, currMaxLength=1

compare first digit of 3 to first digit of 34 → even

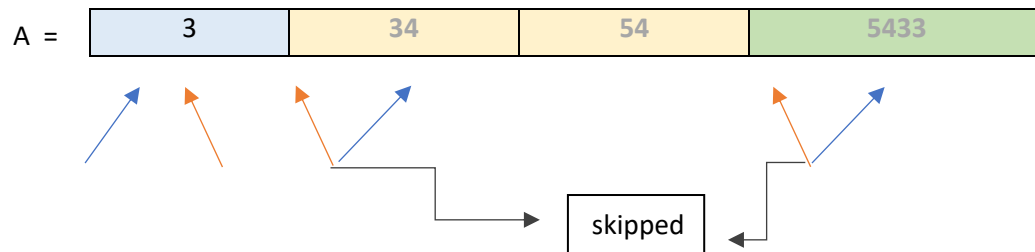
do MSD of 3 = 1

compare first digit of 3 to second digit of 34 → 34 wins

currMax = 34, currMaxLength = 2;

third integer for $r=34$

$r = 54543334$



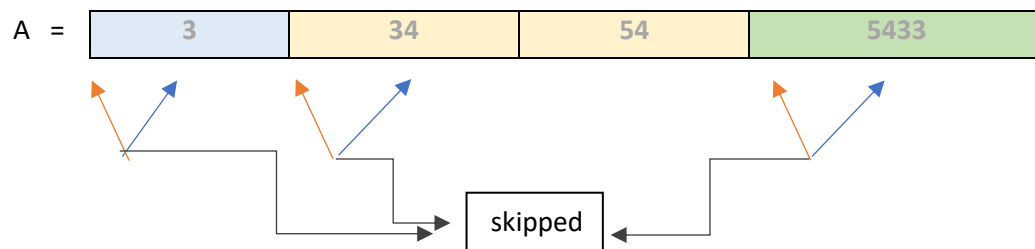
currMax = 0, currMaxLength = 1;

compare first digit of 3 to first digit of 0 → 3 wins

currMax=3, currMaxLength=1

fourth integer for $r=3$

$r = 545433343$



result: $r = 545433343$