2022-01-17 Monday Day 4 : Next Permutation

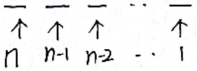
Implement next permutation, which rearranges numbers into the lexicographically next greater permutation of numbers. If such a rearrangement is not possible, it must rearrange it to the lowest possible order. (i.e, sorted in ascending order).

The replacement must be in place and use only constrant extra memory

	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5	Ex 6
INPUT	[1,2,3]	[3,2,1]	[1,1,5]	[4,5,2,6,7,3,1]	[1,5,5]	[5,1,5]
OUTPUT	[1,3,2]	[1,2,3]	[1,5,1]	[4,5,2,7,1,3,6]	[5,1,5]	[5,5,1]

★ Constraints: $1 \leq$ nums.length ≤ 100

$0 \leq$ nums[i] ≤ 100



n! (worst case)

Solution 1: Bruteforce.

Find all permutations: for an n-digit number,

0 1 2 .... n!-1

p(0) p(1) p(2) p(n!-1)

{5,5,1
5,1,5
1,5,5

3.

[1,5,5] – [5,1,5] – [5,5,1]

Find(t) returns index of t:i. If i ≤ n!-2, return list.get(i+1).

Else, return list.get(0)

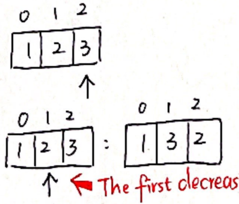
In the worst case, all digits are distincts, the time complexity is O(n!)

Q: Why the space complexity is O(n)? Shouldn't it be O(n!)?

Solution 2. Local SWAP

: We observe that for any given sequence that is in descending order, no next larger permutation is possible. In other words, when the next permutation is possible, there exists some subsequence that is in ascending order.

Ex 1:

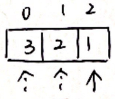


The first decreasing element

i=2, the rightmost element

i=1, stop;

Ex2:



Couldn't find the next decreasing element

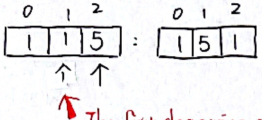
i=2, ...

i=1, continue;

i=0, continue;

i=-1, means the array is already sorted in a non-increasing order.

Ex3:

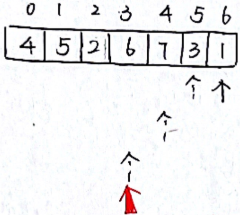


The first decreasing element

i=2,...

i=1, stop.

Ex4:



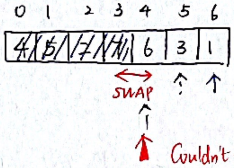
i=6,..

i=5,..

i=4,..

i=3, stop.

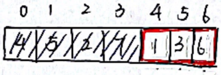
The first decreasing element



The number which is just larger than 6 among {7,3,1}: 7.

Now for indices 4-6, we need the smallest permutation formed by {7,3,1}.

Couldn't find the next decreasing element.



Place {1,3,6} in ascending order to get the smallest permutation

But recall that while scanning the numbers from the right, all numbers to the right of a[3] is already in descending order, and swapping a[3] with a[4] doesn't change the order.

	$k < x \geq y \geq z$		After
Ex1	4 $\underline{6}$ 2 1	$\longrightarrow$	6 4 2 1
Ex2	4 6 $\underline{5}$ 1	SWAP	5 6 4 1
Ex3	4 $\underline{6}$ 4 1		6 4 4 1

Proof: We show that SWAP doesn't change the lexicographical order of elements to the right of a[i-1]

WLOG, let $k < x \geq y \geq z$, and $a[i-1] = k$, $a[j] = \min\{t;\ t \in \{x, y, z\} \text{ and } t > k\}$.

After SWAP, we have a[j] ... t ... We proceed by case distinctions.

Case 1: $a[j] = x$. This means $x > k$ and $y, z \leq k$. After SWAP, we have

x, k, y, z, where $k \geq y \geq z$.

Case 2: $a[j] = y$. This means $y > k$ and $z \leq k$. After SWAP, we have

$y\ x\ k\ z$, where $x \geq y > k \geq z$.

Case 3: $a[j] = z$. This means $z > k$. After SWAP, we have $z \ x \ y \ k$, where

$x \geq y \geq z > k$.

From the three cases above, the local lexicugraphical is preserved after SWAP. since the order before block $[x, y, z]$ to the right of $a'[i-1]$ is preserved, so is the order after block $[x, y, z]$, the order to the right of $a[i-1]$ is preserved.

▨

Therefore, for numbers to the right of the swapped $a[i-1]$ (i.e., $a'[i-1]$), we simply need to reverse the order.

★ Edge Case: Since $1 \leq$ nums.length ≤ 100, when nums.length $= 1$, returns nums.

Ex 5: