

Question: Design an algorithm that takes as input a 9 digit number where no digit appears twice and produces as output an arrangement of the same 9 digits corresponding to the next highest number. If no such number exists, the algorithm should indicate this. So for example if the input is 781623954 the output would be 781624359. You can use bulleted english to describe your algorithm or pseudocode similar to what we saw in class. you do NOT have to write this application in Python. Your algorithm will be graded for efficiency.

Most efficient:

Assuming the number is provided as a list that is indexed from 0
eg: 123 will be [1,2,3]

```
function next_number(num_list):
    i = len(num_list)
    while num_list[i] > num_list[i-1] and i > 0
        i = i - 1
    if i == 0:
        there is no larger number possible
    swap num_list[i] with num_list[i-1]
    sort the numbers in num_list from i until the end
```

To break it down into steps:

- Start from least significant digit (ones).
- Move towards the left, comparing each pair of adjacent values until finding a pair where the left is less than the right ($L < R$).
- If no such pair is found, meaning each $L > R$, then the sequence of digits is already in ascending order right-to-left and is the highest value (987654321)
- If a pair is found such that $L < R$, find the MIN value to the right of L, including R.
- Swap L with MIN
- Sort all values to the left of MIN in descending order, from right to left.

Brute force 1:

```
while number < 987654321:
    if all the digits in the number are unique:
        return number
    number = number + 1
```

Brute force 2:

Compute all permutations of [9,8,7,6,5,4,3,2,1] in order (= 9! Possibilities)
Search and find current permutation
Return next permutation in list (next highest)