task-3

December 18, 2025

[1]:
```python
from typing import List
```

1. Defanging an IP Address

Given a valid (IPv4) IP address, return a defanged version of that IP address.

A defanged IP address replaces every period "." with "[.]".

[2]:
```python
class Solution:
    def defangIPaddr(self, address: str) -> str:
        result = ""
        for ch in address:
            if ch == ".":
                result += "[.]"
            else:
                result += ch
        return result
```

[3]:
```python
class Solution:
    def defangIPaddr(self, address: str) -> str:
        return address.replace(".", "[.]")
```

2. Find Numbers with Even Number of Digits

Given an array nums of integers, return how many of them contain an even number of digits.

[4]:
```python
class Solution:
    def findNumbers(self, nums: List[int]) -> int:
        count = 0
        for num in nums:
            if len(str(num)) % 2 == 0:
                count += 1
        return count
```

[5]:
```python
class Solution:
    def findNumbers(self, nums: List[int]) -> int:
        return sum(1 for num in nums if len(str(num)) % 2 == 0)
```

3. Number of Good Pairs

Given an array of integers nums, return the number of good pairs.

A pair (i, j) is called good if nums[i] == nums[j] and i < j.

[6]:
```python
class Solution:
    def numIdenticalPairs(self, nums: List[int]) -> int:
        freq = {}
        pairs = 0

        for num in nums:
            if num in freq:
                pairs += freq[num]
                freq[num] += 1
            else:
                freq[num] = 1

        return pairs
```

[7]:
```python
class Solution:
    def numIdenticalPairs(self, nums: List[int]) -> int:
        from collections import Counter
        return sum(v * (v - 1) // 2 for v in Counter(nums).values())
```

4. How Many Numbers Are Smaller Than the Current Number

Given the array nums, for each nums[i] find out how many numbers in the array are smaller than it. That is, for each nums[i] you have to count the number of valid j's such that j != i and nums[j] < nums[i].

Return the answer in an array.

[8]:
```python
class Solution:
    def smallerNumbersThanCurrent(self, nums: List[int]) -> List[int]:
        count = [0] * 101

        # frequency count
        for num in nums:
            count[num] += 1

        # prefix sum: count of numbers < i
        for i in range(1, 101):
            count[i] += count[i - 1]

        result = []
        for num in nums:
            if num == 0:
                result.append(0)
            else:
                result.append(count[num - 1])
```

```
        return result
```

[9]:
```
class Solution:
    def smallerNumbersThanCurrent(self, nums: List[int]) -> List[int]:
        sorted_nums = sorted(nums)
        return [sorted_nums.index(num) for num in nums]
```

5. Subtract the Product and Sum of Digits of an Integer

Given an integer number n, return the difference between the product of its digits and the sum of its digits.

[10]:
```
class Solution:
    def subtractProductAndSum(self, n: int) -> int:
        product = 1
        summation = 0

        while n > 0:
            digit = n % 10
            product *= digit
            summation += digit
            n //= 10

        return product - summation
```

[11]:
```
class Solution:
    def subtractProductAndSum(self, n: int) -> int:
        digits = [int(d) for d in str(n)]
        product = 1
        for d in digits:
            product *= d
        return product - sum(digits)
```

6. XOR Operation in an Array

You are given an integer n and an integer start.

Define an array nums where nums[i] = start + 2 * i (0-indexed) and n == nums.length.

Return the bitwise XOR of all elements of nums.

[12]:
```
class Solution:
    def xorOperation(self, n: int, start: int) -> int:
        result = 0
        for i in range(n):
            result ^= start + 2 * i
        return result
```

[13]:
```python
class Solution:
    def xorOperation(self, n: int, start: int) -> int:
        return __import__("functools").reduce(lambda x, y: x ^ y,
               (start + 2*i for i in range(n)), 0)
```