Product of array except self - using brute force

```
class Solution:
    def prod_of_array(self, nums: list[int]) -> list[int]:

        n = len(nums)
        result = [0] * n

        for i in range(n):
            prod = 1
            for j in range(n):
                if i == j:
                    continue
                prod *= nums[j]

            result[i] = prod

        return result

print(Solution().prod_of_array(nums=[1,2,4,6]))
```

Step 1:

self -> solution instance

nums -> [1, 2, 4, 6]

n -> 4

result -> [0, 0, 0, 0]

Step 2:
```
for i in range(n):
    prod = 1
```

i -> 0

prod -> 1

Step 3:
```
for j in range(n):
    if i == j:
        continue
    prod *= nums[j]
```

1st iteration:

j -> 0

i == j True -> continue

2nd iteration:

j -> 1

nums = [1, 2, 4, 6] (0, 1, 2, 3)

prod -> 1 * 2 -> 2

3rd iteration:

j -> 2

prod -> 2 * 4 -> 8

4th iteration:

j -> 3

prod -> 8 * 6 -> 48

Step 4: result[i] = prod

result = [48, 0, 0, 0]

Step 5:

i becomes 1, then prod restarts to the initial value of 1. then j becomes 0 again and prod will be 1 and so on. when j becomes 1, the iteration will then be skipped, then j -> 2 therefore prod -> 4 => j -> 3 and prod -> 24 this will result in having -> [48, 24, 0, 0]

Step 6:

i becomes 2, then prod restarts to the initial value of 1.

then j becomes 0 again and prod will be 1 and so on.
when j becomes 2, the iteration will then be skipped, then
j -> 3 and prod -> 12 =>
result -> [48, 24, 12, 0], after repeating the process
the result will be -> [48, 24, 12, 8]