educative

Solution Review: List of Products of All Elements

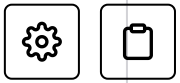
This review provides a detailed analysis of the different ways to find products of all elements in a list.

We'll cover the following

Solution #1: Using a nested loop

```
def find_product(lst):
    result = []
    left = 1  # To store product of all previous values from currentInd
    for i in range(len(lst)):
        currentproduct = 1  # To store current product for index i
        # compute product of values to the right of i index of list
        for ele in lst[i+1:]:
            currentproduct = currentproduct * ele
        # currentproduct * product of all values to the left of i index
        result.append(currentproduct * left)
        # Updating `left`
```

```
12          left = left * lst[i]
13
14      return result
15
16
17  print(find_product([1, 2, 3, 4]))
18
```



This solution iterates over the list and calculates the product of all the numbers to the right of the current element as on **lines 7 and 8**. Then it calculates the product of all the elements to the left of the current element **line 10**. It then multiplies the two products and returns the result **line 14**.

Time Complexity

This algorithm is in $O(n^2)$ because the list is iterated over $n(n-1)/2$ times.

Solution #2: Optimizing the number of multiplications

```
1  def find_product(lst):
2      # get product start from left
3      left = 1
4      product = []
5      for ele in lst:
6          product.append(left)
```

```
6           product.append(left)
7           left = left * ele
8       # get product starting from right
9       right = 1
10      for i in range(len(lst)-1, -1, -1):
11          product[i] = product[i] * right
12          right = right * lst[i]
13
14      return product
15
16
17  print(find_product([0, 1, 2, 3]))
18
```

The algorithm for this solution is to first create a new list with products of all elements to the left of each element as done on **lines 4-6**. Then multiply each element in that list to the product of all the elements to the right of the list by traversing it in reverse as done on **lines 9-11**

Time Complexity

Since this algorithm only traverses over the list twice, it's in linear time, $O(n)$.

Back

Challenge 4: List of Products of all Ele...

Next

Challenge 5: Find Minimum Value in List