



Choose a section

- ☐ Solve
- ☒ Practice

DSA.ai

Interactive Problem Solving

Problem Statement: Given a string of numbers, find the maximum product of three non-overlapping substrings of length 3.

Input/Output Specifications:

- Input: A string of positive integers (digits) separated by spaces.
- Output: The maximum product of three non-overlapping substrings of length 3.

Constraints:

- The input string will only contain digits.
- The length of the input string is at least 6 and at most 100.
- Each substring will have a length of 3.
- No two substrings will overlap.
- All integers in the input string are positive.

Example:

Input: "123456789" Output: 325008

Explanation: The maximum product of three non-overlapping substrings of length 3 is 325008, which is obtained by taking the substrings "123", "234", and "345".

Challenge: This problem requires you to write an efficient algorithm that can find the maximum product of three non-overlapping substrings of length 3 in a given string. The algorithm should take into account the constraints mentioned above, such as no overlap between substrings and all integers being positive.

Write the solution:

sdaasdasasd

Show Solution

Question: Given a string of numbers, find the maximum product of three non-overlapping substrings of length 3.

Classification: Type Classification: String, Difficulty Level: Hard

Python Code Solution:

```
def max_product(s):  
    # Initialize variables to keep track of the maximum product and the indices of the three substrings  
    max_prod = 0  
    i1 = j1 = k1 = 0  
    i2 = j2 = k2 = 0  
    i3 = j3 = k3 = 0  
  
    # Iterate through the string and keep track of the indices of the three substrings  
    for i in range(len(s)):  
        if i - i1 == 3:  
            # If the current index is 3 indices after the first substring, update the second substring's indices  
            i2 = j2 + 1  
            j2 = k2 + 1  
            k2 = i  
        elif i - i2 == 3:  
            # If the current index is 3 indices after the second substring, update the third substring's indices  
            i3 = j3 + 1  
            j3 = k3 + 1  
            k3 = i  
  
        # Update the maximum product based on the current substring's product  
        prod = int(s[i1:j1+1]) * int(s[i2:j2+1]) * int(s[i3:j3+1])  
        if prod > max_prod:  
            max_prod = prod
```

```
# Return the maximum product
return max_prod
```

Summary: The algorithm iterates through the string and keeps track of the indices of the three substrings. For each index, it updates the indices of the second and third substrings based on whether they are 3 indices after the first substring or not. It then calculates the product of the current substring with the previous two substrings and updates the maximum product accordingly.

Hints:

- To calculate the product of three integers, you can multiply them together using the `*` operator.
- To convert a string to an integer, you can use the `int()` function.
- To determine whether an index is 3 indices after another index, you can subtract the two indexes and check if the result is equal to 3.

Similar Questions:

- Find the maximum product of three non-overlapping substrings of length 3 in a given string.
- Given a string of positive integers, find the maximum product of three non-overlapping substrings of length 3.
- Given a string of digits, find the maximum product of three consecutive substrings of length 3.