Integer Multiplication:

The integer multiplication problem is actually one of the simplest mathematical problems. It consists of printing the result obtained by multiplying two integers, but we wanted to take this problem one step further and be more efficient. The reason for this was that the number of multiplications increased greatly when doing large integer multiplication.

```
     456              456
 ×  1234          ×  1234
 ---------        ----------
    1824              456
  + 1368           + 0912
 + 0912            +  1368
 0456              +   1824
 -------------    -------------
   562704           562704
  American           British
```

As seen in the example on the left, each number is multiplied and added together. This algorithm takes $O(n^2)$ time.

Karatsuba Multiplication, which uses the divide and conquer management, is used exactly here. Here's a more detailed explanation of the problem:

Input:

Integers that we can call A and B

Then, two numbers are divided in half and converted into numbers that we can name as a, b, c and d.

Output:

The output of this problem is the result of efficiently multiplying two inputs using algorithm.

Problem Statement:

This problem arose due to the inefficiency of the process solved by the traditional method. With Karatsuba Multiplication, we divide the multiplication into sub-problems and get more efficient results in a recursive way.

1.Divide: The process is performed by recursively dividing the numbers A and B specified in the input by $n/2$ digits until the required size is reached.

2.Conquer: P1: Multiplication of b and d; P2: Multiplication of a and c;

P3: Multiplication of (b + a) and (c + d)

3.Merge: Result = $(P1 * 10^n) + ((P3 - P1 - P2) * 10^{(n/2)}) + P2$

Time Cost:

With the Karatsuba Multiplication algorithm, a greater efficiency is achieved compared to the method shown at the beginning of the article. While the time cost of the traditional model is $O(n^2)$ time, the Karatsuba Multiplication algorithm is $O(n^{\log_2(3)})$ time. The reason for this is this:

As mentioned above, the algorithm performs $n/2$ size integers by dividing them into 3 subproblems. In this way, time cost becomes $O(n^{\log_2(3)})$. This is approximately equal to $O(n^{1.59})$.

Space Cost:

While the space cost of traditional multiplication is $O(n)$, it is reduced to $O(\log n)$ with the Karatsuba algorithm. To explain in detail:

The algorithm is recursive. At each step, the operation is carried out by dividing the digits of the numbers in half until the operation is completed. This is equal to $\log_2 n$. In terms of space cost, it is equal to $O(\log n)$.