Suppose we have $Input = [a_0, a_1, a_i, \ldots, a_n]$, and we also define
tmp_i = the total strength of all sub arrays ends at i.
For example, tmp_1 = totoal strength of $[a_0, a_1]$+ total strength of $[a_1]$.
Thus, Answer = $\sum_{i=0}^{i=n} tmp_i$
We also maintain a monotonic stack $S = [s_0, s_1, \ldots, s_i]$
Each s_i is an index of input, where $a_{s_0} < a_{s_1} < \ldots < a_{s_i}$
We need define two helper functions.
$sum(l, r) = \sum_{i=l}^{i=r} a_i$
$f(l, r) = \sum_{i=l}^{i=r} sum(i, r)$
Suppose at this moment, there's $[s_0, s_1, \ldots, s_i]$

$$\begin{aligned} tmp_i = & \ sum(0, s_i) * a_{s_0} + \ldots + (a_{s_0} + sum(s_0 + 1, s_i)) * a_{s_0} \\ & + sum(s_0 + 1, s_i) * a_{s_1} + \ldots + (a_{s_1} + sum(s_1 + 1, s_i)) * a_{s_1} \\ & + \ldots \\ & + sum(s_{i-1} + 1, s_i) * a_{s_i} + \ldots + a_{s_i} * a_{s_i} \end{aligned} \tag{1}$$

Then we add a_{s_i+1} into the stack S. Now:

$$\begin{aligned} tmp_{i+1} = & \ sum(0, s_{i+1}) * a_{s_0} + \ldots + (a_{s_0} + sum(s_0 + 1, s_{i+1})) * a_{s_0} \\ & + sum(s_0 + 1, s_{i+1}) * a_{s_1} + \ldots + (a_{s_1} + sum(s_1 + 1, s_{i+1})) * a_{s_1} \\ & + \ldots \\ & + sum(s_{i-1} + 1, s_{i+1}) * a_{s_i} + \ldots + (a_{s_i} + sum(s_i + 1, s_{i+1})) * a_{s_i} \\ & + sum(s_i + 1, s_{i+1}) * a_{s_{i+1}} + \ldots + a_{s_{i+1}} * a_{s_{i+1}} \end{aligned} \tag{2}$$

If we observe the second equation carefully. We can find.

$$\begin{aligned} tmp_{i+1} = & \ (sum(0, s_i) + sum(s_i + 1, s_{i+1})) * a_{s_0} + \ldots + (a_{s_0} + sum(s_0 + 1, s_i) + sum(s_i + 1, s_{i+1})) * a_{s_0} \\ & + (sum(s_0 + 1, s_i) + sum(s_i + 1, s_{i+1})) * a_{s_1} + \ldots + (a_{s_1} + sum(s_1 + 1, s_i) + sum(s_i + 1, s_{i+1})) * a_{s_1} \\ & + \ldots \\ & + (sum(s_{i-1} + 1, s_i) + sum(s_i + 1, s_{i+1})) * a_{s_i} + \ldots + (a_{s_i} + sum(s_i + 1, s_{i+1})) * a_{s_i} \\ & + sum(s_i + 1, s_{i+1}) * a_{s_{i+1}} + \ldots + a_{s_{i+1}} * a_{s_{i+1}} \end{aligned} \tag{3}$$

And if we manage the equation above, we can get the next equation.

$$\begin{aligned} tmp_{i+1} = & \ sum(0, s_i) * a_{s_0} + \ldots + (a_{s_0} + sum(s_0 + 1, s_i)) * a_{s_0} + sum(s_i + 1, s_{i+1}) * (s_0 + 1) * a_{s_0} \\ & + sum(s_0 + 1, s_i) * a_{s_1} + \ldots + (a_{s_1} + sum(s_1 + 1, s_i)) * a_{s_1} + sum(s_i + 1, s_{i+1}) * (s_1 - s_0) * a_{s_1} \\ & + \ldots \\ & + sum(s_{i-1} + 1, s_i) * a_{s_i} + \ldots + a_{s_i} * a_{s_i} + sum(s_i + 1, s_{i+1}) * (s_i - s_{i-1}) * a_{s_i} \\ & + sum(s_i + 1, s_{i+1}) * a_{s_{i+1}} + \ldots + a_{s_{i+1}} * a_{s_{i+1}} \end{aligned} \tag{4}$$

And we can re-write it as

$$\begin{aligned} tmp_{i+1} = & \ tmp_i \\ & + sum(s_i + 1, s_{i+1}) * (s_0 + 1) * a_{s_0} \\ & + sum(s_i + 1, s_{i+1}) * (s_1 - s_0) * a_{s_1} \\ & + \ldots \\ & + sum(s_i + 1, s_{i+1}) * (s_i - s_{i-1}) * a_{s_i} \\ & + sum(s_i + 1, s_{i+1}) * a_{s_{i+1}} + \ldots + a_{s_{i+1}} * a_{s_{i+1}} \end{aligned} \tag{5}$$

We can then define V_i

$$\begin{aligned} V_i = & \ (s_0 + 1) * a_{s_0} \\ & + (s_1 - s_0) * a_{s_1} \\ & + \ldots \\ & + (s_i - s_{i-1}) * a_{s_i} \end{aligned} \tag{6}$$

And for V_{i+1}

$$\begin{aligned} V_{i+1} = & (s_0 + 1) * a_{s_0} \\ & + (s_1 - s_0) * a_{s_1} \\ & + \ldots \\ & + (s_i - s_{i-1}) * a_{s_i} \\ & + (s_{i+1} - s_i) * a_{s_{i+1}} \\ & = V_i + (s_{i+1} - s_i) * a_{s_{i+1}} \end{aligned} \tag{7}$$

And then, we can rewrite equation 5 to

$$\begin{aligned} tmp_{i+1} = & tmp_i \\ & + V_i * sum(s_i + 1, s_{i+1}) \\ & + sum(s_i + 1, s_{i+1}) * a_{s_{i+1}} + \ldots + a_{s_{i+1}} * a_{s_{i+1}} \\ & = tmp_i + V_i * sum(s_i + 1, s_{i+1}) + f(s_i + 1, s_{i+1}) * a_{s_{i+1}} \end{aligned} \tag{8}$$

Wow! we finally got the equation to anser tmp_i!

Please note, we need to make S as a monotonic stack, so when we pop element, we need to reverse tmp and V according to the equation above as well.

And to make $f(l, r)$ be a O(1) operation, we need to compute the suffix sum of prefix sum. I think the code explains it pretty well.