Problem 1039: Minimum Score Triangulation of Polygon

Problem Information

Difficulty: Medium
Acceptance Rate: 67.28%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

You have a convex `n`-sided polygon where each vertex has an integer value. You are given an integer array `values` where `values[i]` is the value of the `ith` vertex in **clockwise order**.

Polygon **triangulation** is a process where you divide a polygon into a set of triangles and the vertices of each triangle must also be vertices of the original polygon. Note that no other shapes other than triangles are allowed in the division. This process will result in `n - 2` triangles.

You will **triangulate** the polygon. For each triangle, the _weight_ of that triangle is the product of the values at its vertices. The total score of the triangulation is the sum of these _weights_ over all `n - 2` triangles.

Return the _minimum possible score_ that you can achieve with some __**triangulation** __ of the polygon.

Example 1:

Input: values = [1,2,3]

Output: 6

Explanation: The polygon is already triangulated, and the score of the only triangle is 6.

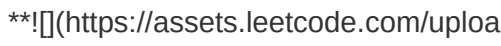
Example 2:

Input: values = [3,7,4,5]

Output: 144

Explanation: There are two triangulations, with possible scores: 3*7*5 + 4*5*7 = 245, or 3*4*5 + 3*4*7 = 144. The minimum score is 144.

Example 3:

**  ■■■■■■■**

Input: values = [1,3,1,4,1,5]

Output: 13

Explanation: The minimum score triangulation is 1*1*3 + 1*1*4 + 1*1*5 + 1*1*1 = 13.

Constraints:

* `n == values.length` * `3 <= n <= 50` * `1 <= values[i] <= 100`

Code Snippets

C++:

```
class Solution {
public:
int minScoreTriangulation(vector<int>& values) {

}
};
```

Java:

```
class Solution {
public int minScoreTriangulation(int[] values) {

}
}
```

Python3:

```
class Solution:
def minScoreTriangulation(self, values: List[int]) -> int:
```