

Word Wrap

Given an array **nums[]** of size **n**, where **nums[i]** denotes the number of characters in one word. Let **K** be the limit on the number of characters that can be put in one line (line width). Put line breaks in the given sequence such that the lines are printed neatly.

Assume that the length of each word is smaller than the line width. When line breaks are inserted there is a possibility that extra spaces are present in each line. The extra spaces include spaces put at the end of every line **except the last one**.

You have to **minimize** the following total cost where **total cost** = Sum of cost of all lines, where cost of line is = (Number of extra spaces in the line)².

Example 1:

Input: nums = {3,2,2,5}, k = 6

Output: 10

Explanation: Given a line can have 6 characters,

Line number 1: From word no. 1 to 1

Line number 2: From word no. 2 to 3

Line number 3: From word no. 4 to 4

So total cost = $(6-3)^2 + (6-2-2-1)^2 = 3^2 + 1^2 = 10$.

As in the first line word length = 3 thus

extra spaces = $6 - 3 = 3$ and in the second line

there are two word of length 2 and there already

1 space between two word thus extra spaces

= $6 - 2 - 2 - 1 = 1$. As mentioned in the problem

description there will be no extra spaces in

the last line. Placing first and second word in first line and third word on second line would take a cost of $0^2 + 4^2 = 16$ (zero spaces on first line and $6-2 = 4$ spaces on second), which isn't the minimum possible cost.

Example 2:

Input: `nums = {3,2,2}`, `k = 4`

Output: 5

Explanation: Given a line can have 4 characters,

Line number 1: From word no. 1 to 1

Line number 2: From word no. 2 to 2

Line number 3: From word no. 3 to 3

Same explanation as above total cost
 $= (4 - 3)^2 + (4 - 2)^2 = 5$.

Your Task:

You don't need to read or print anything. Your task is to complete the function **`solveWordWrap()`** which takes `nums` and `k` as input parameter and returns the minimized total cost.

Expected Time Complexity: $O(n^2)$

Expected Space Complexity: $O(n)$

Constraints:

$1 \leq n \leq 500$

$1 \leq \text{nums}[i] \leq 1000$

$\max(\text{nums}[i]) \leq k \leq 2000$