You are given a string text. We want to display text on a screen of width w and height h. You can choose any font size from array fonts, which contains the available font sizes **in ascending order**.

You can use the FontInfo interface to get the width and height of any character at any available font size.

The FontInfo interface is defined as such:

interface FontInfo {

// Returns the width of character ch on the screen using font size fontSize.

// O(1) per call

public int getWidth(int fontSize, char ch);

// Returns the height of any character on the screen using font size fontSize.

// O(1) per call

public int getHeight(int fontSize);

}

The calculated width of text for some fontSize is the **sum** of every getWidth(fontSize, text[i]) call for each 0 <= i < text.length (**0-indexed**). The calculated height of text for some fontSize is getHeight(fontSize). Note that text is displayed on a **single line**.

It is guaranteed that FontInfo will return the same value if you call getHeight or getWidth with the same parameters.

It is also guaranteed that for any font size fontSize and any character ch:

* getHeight(fontSize) <= getHeight(fontSize+1)
* getWidth(fontSize, ch) <= getWidth(fontSize+1, ch)

Return *the maximum font size you can use to display*text*on the screen*. If text cannot fit on the display with any font size, return -1.

**Example 1:**

**Input:** text = "helloworld", w = 80, h = 20, fonts = [6,8,10,12,14,16,18,24,36]

**Output:** 6

**Example 2:**

**Input:** text = "leetcode", w = 1000, h = 50, fonts = [1,2,4]

**Output:** 4

**Example 3:**

**Input:** text = "easyquestion", w = 100, h = 100, fonts = [10,15,20,25]

**Output:** -1

**Constraints:**

* 1 <= text.length <= 50000
* text contains only lowercase English letters.
* 1 <= w <= 107
* 1 <= h <= 104
* 1 <= fonts.length <= 105
* 1 <= fonts[i] <= 105
* fonts is sorted in ascending order and does not contain duplicates.