Problem Statement

You are working as a software engineer at a financial technology company that specializes in secure and intelligent payment processing solutions. One of your company's newest offerings is an AI-powered transaction analyser, which is designed to streamline digital payment validations based on specific rules and patterns. The objective is to ensure that certain transaction values are aligned with regulatory or promotional conditions — one such condition being divisibility by 25.
In financial systems, values divisible by 25 are often used due to the denominations of many currencies (e.g., quarters, certain fees, or discount structures). Your task is to develop a utility function for a core module that ensures a transaction amount (represented as a string) can be modified to satisfy this divisibility rule by deleting the fewest number of digits. Deleting a digit is treated as an "operation," and minimizing these operations is critical to preserve as much of the original amount as possible.
You are given a non-negative integer represented as a string num. In a single operation, you can remove any one digit from num. You can perform this operation any number of times, but not more than the length of the number. If all digits are removed, the resulting number is considered 0.
An integer is considered special if it is divisible by 25. Your task is to write a function that returns the minimum number of digit deletions (operations) required to make the number special. If it's not possible to make it special through any number of deletions, return the length of the number (i.e., delete all digits to get 0).

Constraints:
1 <= num.length <= 100
num only consists of digits '0' through '9'.
num does not contain any leading zeros

Example 1:
Input: num = "2245047"
Output: 2
Explanation: Delete digits num[5] and num[6]. The resulting number is "22450" which is special since it is divisible by 25.
It can be shown that 2 is the minimum number of operations required to get a special number.

Example 2:
Input: num = "2908305"
Output: 3
Explanation: Delete digits num[3], num[4], and num[6]. The resulting number is "2900" which is special since it is divisible by 25.
It can be shown that 3 is the minimum number of operations required to get a special number.

Example 3:
Input: num = "10"
Output: 1
Explanation: Delete digit num[0]. The resulting number is "0" which is special since it is divisible by 25.
It can be shown that 1 is the minimum number of operations required to get a special number.

Input0:
290830545
Output0:
5

Input1:
345678
Output1:
6

Input2:
2345890
Output2:
2

Input3:
67890
Output3:
4

Input4:
98570
Output4:
1

Input5:
234098
Output5:.
5