Problem 482: License Key Formatting

Problem Information

Difficulty: Easy
Acceptance Rate: 45.16%
Paid Only: No
Tags: String

Problem Description

You are given a license key represented as a string `s` that consists of only alphanumeric characters and dashes. The string is separated into `n + 1` groups by `n` dashes. You are also given an integer `k`.

We want to reformat the string `s` such that each group contains exactly `k` characters, except for the first group, which could be shorter than `k` but still must contain at least one character. Furthermore, there must be a dash inserted between two groups, and you should convert all lowercase letters to uppercase.

Return _the reformatted license key_.

Example 1:

Input: s = "5F3Z-2e-9-w", k = 4 **Output:** "5F3Z-2E9W" **Explanation:** The string s has been split into two parts, each part has 4 characters. Note that the two extra dashes are not needed and can be removed.

Example 2:

Input: s = "2-5g-3-J", k = 2 **Output:** "2-5G-3J" **Explanation:** The string s has been split into three parts, each part has 2 characters except the first part as it could be shorter as mentioned above.

Constraints:

* `1 <= s.length <= 105` * `s` consists of English letters, digits, and dashes `'-'`. * `1 <= k <= 104`

Code Snippets

C++:

```
class Solution {
public:
string licenseKeyFormatting(string s, int k) {

}
};
```

Java:

```
class Solution {
public String licenseKeyFormatting(String s, int k) {

}
}
```

Python3:

```
class Solution:
def licenseKeyFormatting(self, s: str, k: int) -> str:
```