Problem 2370: Longest Ideal Subsequence

Problem Information

Difficulty: Medium
Acceptance Rate: 46.69%
Paid Only: No
Tags: Hash Table, String, Dynamic Programming

Problem Description

You are given a string `s` consisting of lowercase letters and an integer `k`. We call a string `t` **ideal** if the following conditions are satisfied:

* `t` is a **subsequence** of the string `s`. * The absolute difference in the alphabet order of every two **adjacent** letters in `t` is less than or equal to `k`.

Return _the length of the**longest** ideal string_.

A **subsequence** is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters.

Note that the alphabet order is not cyclic. For example, the absolute difference in the alphabet order of `'a'` and `'z'` is `25`, not `1`.

Example 1:

Input: s = "acfgbd", k = 2 **Output:** 4 **Explanation:** The longest ideal string is "acbd". The length of this string is 4, so 4 is returned. Note that "acfgbd" is not ideal because 'c' and 'f' have a difference of 3 in alphabet order.

Example 2:

Input: s = "abcd", k = 3 **Output:** 4 **Explanation:** The longest ideal string is "abcd". The length of this string is 4, so 4 is returned.

Constraints:

* `1 <= s.length <= 105` * `0 <= k <= 25` * `s` consists of lowercase English letters.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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