Problem 3307: Find the K-th Character in String Game II

Problem Information

Difficulty: Hard
Acceptance Rate: 48.62%
Paid Only: No
Tags: Math, Bit Manipulation, Recursion

Problem Description

Alice and Bob are playing a game. Initially, Alice has a string `word = "a"`.

You are given a **positive** integer `k`. You are also given an integer array `operations`, where `operations[i]` represents the **type** of the `ith` operation.

Now Bob will ask Alice to perform **all** operations in sequence:

* If `operations[i] == 0`, **append** a copy of `word` to itself. * If `operations[i] == 1`, generate a new string by **changing** each character in `word` to its **next** character in the English alphabet, and **append** it to the _original_ `word`. For example, performing the operation on `"c"` generates `"cd"` and performing the operation on `"zb"` generates `"zbac"`.

Return the value of the `kth` character in `word` after performing all the operations.

Note that the character `'z'` can be changed to `'a'` in the second type of operation.

Example 1:

Input: k = 5, operations = [0,0,0]

Output: "a"

Explanation:

Initially, `word == "a"`. Alice performs the three operations as follows:

* Appends `"a"` to `"a"`, `word` becomes `"aa"`. * Appends `"aa"` to `"aa"`, `word` becomes `"aaaa"`. * Appends `"aaaa"` to `"aaaa"`, `word` becomes `"aaaaaaaa"`.

Example 2:

Input: k = 10, operations = [0,1,0,1]

Output: "b"

Explanation:

Initially, `word == "a"`. Alice performs the four operations as follows:

* Appends `"a"` to `"a"`, `word` becomes `"aa"`. * Appends `"bb"` to `"aa"`, `word` becomes `"aabb"`. * Appends `"aabb"` to `"aabb"`, `word` becomes `"aabbaabb"`. * Appends `"bbccbbcc"` to `"aabbaabb"`, `word` becomes `"aabbaabbbbccbbcc"`.

Constraints:

* `1 <= k <= 1014` * `1 <= operations.length <= 100` * `operations[i]` is either 0 or 1. * The input is generated such that `word` has **at least** `k` characters after all operations.

Code Snippets

C++:

```
class Solution {
public:
char kthCharacter(long long k, vector<int>& operations) {

}
};
```

Java:

```
class Solution {
public char kthCharacter(long k, int[] operations) {
```

```
}
}
```

Python3:

```
class Solution:
def kthCharacter(self, k: int, operations: List[int]) -> str:
```