Problem 2868: The Wording Game

Problem Information

Difficulty: Hard
Acceptance Rate: 55.14%
Paid Only: Yes
Tags: Array, Math, Two Pointers, String, Greedy, Game Theory

Problem Description

Alice and Bob each have a **lexicographically sorted** array of strings named `a` and `b` respectively.

They are playing a wording game with the following rules:

* On each turn, the current player should play a word from their list such that the new word is **closely greater** than the last played word; then it's the other player's turn. * If a player can't play a word on their turn, they lose.

Alice starts the game by playing her **lexicographically****smallest** word.

Given `a` and `b`, return `true` _if Alice can win knowing that both players play their best, and_ `false` _otherwise._

A word `w` is **closely greater** than a word `z` if the following conditions are met:

* `w` is **lexicographically greater** than `z`. * If `w1` is the first letter of `w` and `z1` is the first letter of `z`, `w1` should either be **equal** to `z1` or be the **letter after** `z1` in the alphabet. * For example, the word `"care"` is closely greater than `"book"` and `"car"`, but is not closely greater than `"ant"` or `"cook"`.

A string `s` is **lexicographically****greater** than a string `t` if in the first position where `s` and `t` differ, string `s` has a letter that appears later in the alphabet than the corresponding letter in `t`. If the first `min(s.length, t.length)` characters do not differ, then the longer string is the lexicographically greater one.

Example 1:

Input: a = ["avokado","dabar"], b = ["brazil"] **Output:** false **Explanation:** Alice must start the game by playing the word "avokado" since it's her smallest word, then Bob plays his only word, "brazil", which he can play because its first letter, 'b', is the letter after Alice's word's first letter, 'a'. Alice can't play a word since the first letter of the only word left is not equal to 'b' or the letter after 'b', 'c'. So, Alice loses, and the game ends.

Example 2:

Input: a = ["ananas","atlas","banana"], b = ["albatros","cikla","nogomet"] **Output:** true **Explanation:** Alice must start the game by playing the word "ananas". Bob can't play a word since the only word he has that starts with the letter 'a' or 'b' is "albatros", which is smaller than Alice's word. So Alice wins, and the game ends.

Example 3:

Input: a = ["hrvatska","zastava"], b = ["bijeli","galeb"] **Output:** true **Explanation:** Alice must start the game by playing the word "hrvatska". Bob can't play a word since the first letter of both of his words are smaller than the first letter of Alice's word, 'h'. So Alice wins, and the game ends.

Constraints:

* `1 <= a.length, b.length <= 105` * `a[i]` and `b[i]` consist only of lowercase English letters. * `a` and `b` are **lexicographically sorted**. * All the words in `a` and `b` combined are **distinct**. * The sum of the lengths of all the words in `a` and `b` combined does not exceed `106`.

Code Snippets

C++:

```
class Solution {
public:
bool canAliceWin(vector<string>& a, vector<string>& b) {

}
};
```

Java:

```
class Solution {
public boolean canAliceWin(String[] a, String[] b) {

}
}
```

Python3:

```
class Solution:
def canAliceWin(self, a: List[str], b: List[str]) -> bool:
```