2573. Find the String with LCP

Description

We define the `lcp` matrix of any **0-indexed** string `word` of `n` lowercase English letters as an `n x n` grid such that:

- `lcp[i][j]` is equal to the length of the **longest common prefix** between the substrings `word[i,n-1]` and `word[j,n-1]`.

Given an `n x n` matrix `lcp`, return the alphabetically smallest string `word` that corresponds to `lcp`. If there is no such string, return an empty string.

A string `a` is lexicographically smaller than a string `b` (of the same length) if in the first position where `a` and `b` differ, string `a` has a letter that appears earlier in the alphabet than the corresponding letter in `b`. For example, `"aabd"` is lexicographically smaller than `"aaca"` because the first position they differ is at the third letter, and `'b'` comes before `'c'`.

Example 1:

```
Input: lcp = [[4,0,2,0],[0,3,0,1],[2,0,2,0],[0,1,0,1]]
Output: "abab"
Explanation: lcp corresponds to any 4 letter string with two alternating letters. The lexicographically smallest of them is "abab".
```

Example 2:

```
Input: lcp = [[4,3,2,1],[3,3,2,1],[2,2,2,1],[1,1,1,1]]
Output: "aaaa"
Explanation: lcp corresponds to any 4 letter string with a single distinct letter. The lexicographically smallest of them is "aaaa".
```

Example 3:

```
Input: lcp = [[4,3,2,1],[3,3,2,1],[2,2,2,1],[1,1,1,3]]
Output: ""
Explanation: lcp[3][3] cannot be equal to 3 since word[3,...,3] consists of only a single letter; Thus, no answer exists.
```

Constraints:

- `1 <= n ==` `lcp.length ==` `lcp[i].length` `<= 1000`
- `0 <= lcp[i][j] <= n`