Problem 3720: Lexicographically Smallest Permutation Greater Than Target

Problem Information

Difficulty: Medium
Acceptance Rate: 25.80%
Paid Only: No
Tags: Hash Table, String, Greedy, Counting, Enumeration

Problem Description

You are given two strings `s` and `target`, both having length `n`, consisting of lowercase English letters.

Return the **lexicographically smallest permutation** of `s` that is **strictly** greater than `target`. If no permutation of `s` is lexicographically strictly greater than `target`, return an empty string.

A string `a` is **lexicographically strictly greater** 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 later in the alphabet than the corresponding letter in `b`.

Example 1:

Input: s = "abc", target = "bba"

Output: "bca"

Explanation:

* The permutations of `s` (in lexicographical order) are `"abc"`, `"acb"`, `"bac"`, `"bca"`, `"cab"`, and `"cba"`. * The lexicographically smallest permutation that is strictly greater than `target` is `"bca"`.

Example 2:

Input: s = "leet", target = "code"

Output: "eelt"

Explanation:

* The permutations of `s` (in lexicographical order) are `"eelt"`, `"eetl"`, `"elet"`, `"elte"`, `"etel"`, `"etle"`, `"leet"`, `"lete"`, `"ltee"`, `"teel"`, `"tele"`, and `"tlee"`. * The lexicographically smallest permutation that is strictly greater than `target` is `"eelt"`.

Example 3:

Input: s = "baba", target = "bbaa"

Output: ""

Explanation:

* The permutations of `s` (in lexicographical order) are `"aabb"`, `"abab"`, `"abba"`, `"baab"`, `"baba"`, and `"bbaa"`. * None of them is lexicographically strictly greater than `target`. Therefore, the answer is `""`.

Constraints:

* `1 <= s.length == target.length <= 300` * `s` and `target` consist of only lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
string lexGreaterPermutation(string s, string target) {

}
};
```

Java:

```
class Solution {
public String lexGreaterPermutation(String s, String target) {

}
}
```

Python3:

```
class Solution:
def lexGreaterPermutation(self, s: str, target: str) -> str:
```