Problem 3518: Smallest Palindromic Rearrangement II

Problem Information

Difficulty: Hard
Acceptance Rate: 13.48%
Paid Only: No
Tags: Hash Table, Math, String, Combinatorics, Counting

Problem Description

You are given a **palindromic** string `s` and an integer `k`.

Return the **k-th** **lexicographically smallest** palindromic permutation of `s`. If there are fewer than `k` distinct palindromic permutations, return an empty string.

Note: Different rearrangements that yield the same palindromic string are considered identical and are counted once.

Example 1:

Input: s = "abba", k = 2

Output: "baab"

Explanation:

* The two distinct palindromic rearrangements of `"abba"` are `"abba"` and `"baab"`. * Lexicographically, `"abba"` comes before `"baab"`. Since `k = 2`, the output is `"baab"`.

Example 2:

Input: s = "aa", k = 2

Output: ""

Explanation:

* There is only one palindromic rearrangement: `"aa"`. * The output is an empty string since `k = 2` exceeds the number of possible rearrangements.

Example 3:

Input: s = "bacab", k = 1

Output: "abcba"

Explanation:

* The two distinct palindromic rearrangements of `"bacab"` are `"abcba"` and `"bacab"`. * Lexicographically, `"abcba"` comes before `"bacab"`. Since `k = 1`, the output is `"abcba"`.

Constraints:

* `1 <= s.length <= 104` * `s` consists of lowercase English letters. * `s` is guaranteed to be palindromic. * `1 <= k <= 106`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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