Problem 1850: Minimum Adjacent Swaps to Reach the Kth Smallest Number

Problem Information

Difficulty: Medium
Acceptance Rate: 71.76%
Paid Only: No
Tags: Two Pointers, String, Greedy

Problem Description

You are given a string `num`, representing a large integer, and an integer `k`.

We call some integer **wonderful** if it is a **permutation** of the digits in `num` and is **greater in value** than `num`. There can be many wonderful integers. However, we only care about the **smallest-valued** ones.

* For example, when `num = "5489355142"`: * The 1st smallest wonderful integer is `"5489355214"`. * The 2nd smallest wonderful integer is `"5489355241"`. * The 3rd smallest wonderful integer is `"5489355412"`. * The 4th smallest wonderful integer is `"5489355421"`.

Return _the**minimum number of adjacent digit swaps** that needs to be applied to _`num` _to reach the_`kth` _**smallest wonderful** integer_.

The tests are generated in such a way that `kth` smallest wonderful integer exists.

Example 1:

Input: num = "5489355142", k = 4 **Output:** 2 **Explanation:** The 4th smallest wonderful number is "5489355421". To get this number: - Swap index 7 with index 8: "5489355 _14_ 2" -> "5489355 _41_ 2" - Swap index 8 with index 9: "54893554 _12_ " -> "54893554 _21_ "

Example 2:

Input: num = "11112", k = 4 **Output:** 4 **Explanation:** The 4th smallest wonderful number is "21111". To get this number: - Swap index 3 with index 4: "111 _12_ " -> "111 _21_ " - Swap index 2 with index 3: "11 _12_ 1" -> "11 _21_ 1" - Swap index 1 with index 2: "1 _12_ 11" -> "1 _21_ 11" - Swap index 0 with index 1: "_12_ 111" -> "_21_ 111"

Example 3:

Input: num = "00123", k = 1 **Output:** 1 **Explanation:** The 1st smallest wonderful number is "00132". To get this number: - Swap index 3 with index 4: "001 _23_ " -> "001 _32_ "

Constraints:

* `2 <= num.length <= 1000` * `1 <= k <= 1000` * `num` only consists of digits.

Code Snippets

C++:

```
class Solution {
public:
int getMinSwaps(string num, int k) {

}
};
```

Java:

```
class Solution {
public int getMinSwaps(String num, int k) {

}
}
```

Python3:

```
class Solution:
def getMinSwaps(self, num: str, k: int) -> int:
```