Documentation: Distinct Subsequences

Problem Statement

Given two strings, s and t, the goal is to determine the number of distinct subsequences of s that equal t. The problem ensures that the answer fits within a 32-bit signed integer.

Example 1

- **Input:**
  - s = "rabbbit"
  - t = "rabbit"
- **Output:** 3
- **Explanation:**
  - *There are 3 ways to generate the string "rabbit" from "rabbbit":*
    - rabbbit
    - rabbbit
    - rabbit

Example 2

- **Input:**
  - s = "babgbag"
  - t = "bag"
- **Output:** 5
- **Explanation:**
  - *There are 5 ways to generate the string "bag" from "babgbag":*
    - babgbag
    - babgbag
    - babgbag
    - babgbag
    - babgbag

Constraints

- The length of s and t are between 1 and 1000 inclusive.
- Both s and t consist of English letters only.

Approach

To solve the problem of finding the number of distinct subsequences of s that equal t, we can use dynamic programming. The main idea is to use a 2D array dp where dp[i][j] represents the number of distinct subsequences of the first i characters of s that equal the first j characters of t.

Steps

1. **Initialization:**

- Define the lengths of the strings s and t as m and n, respectively.
- Create a 2D array dp with dimensions (m+1) x (n+1), initialized to 0.
- Initialize the first column of dp to 1, since an empty t is a subsequence of any prefix of s.

2. **Filling the DP Table:**

- Iterate over the characters of s and t using two nested loops.
- *For each pair of indices (i, j), update the dp table based on the following conditions:*
  - If the characters s[i-1] and t[j-1] are equal, then dp[i][j] is the sum of dp[i-1][j] and dp[i-1][j-1].
  - If the characters are not equal, then dp[i][j] is equal to dp[i-1][j].

3. **Result:**

- The final answer, representing the number of distinct subsequences of s that equal t, is stored in dp[m][n].

Time and Space Complexity

- **Time Complexity:** O(m * n), where m is the length of s and n is the length of t, since we need to fill a 2D array of size (m+1) x (n+1).
- **Space Complexity:** O(m * n), due to the space required to store the dp table.