

Intuition

just explore all the option -> take if current character is same with any of the string
current character string 1 and string 2;

Approach

1. The solve function is a recursive helper function that takes the current indices ind1 and ind2 for strings s1 and s2 respectively, along with a memoization table dp. The purpose of this function is to check if it's possible to create the remaining part of s3 (starting from ind1+ind2 position) using the remaining parts of s1 (starting from ind1 position) and s2 (starting from ind2 position).
2. The base case for the recursion is when the sum of ind1 and ind2 equals the length of s3, meaning all characters of s3 have been matched successfully. In this case, the function returns true.
3. Before proceeding with the actual computation, the function checks if the result for the current ind1 and ind2 indices has already been computed and stored in the memoization table dp. If so, it returns the precomputed result.
4. The function initializes a boolean variable ans to false. It then checks two conditions:
 - If ind1 is within bounds of s1 and the character at s1[ind1] matches the character at s3[ind1+ind2], it recursively calls solve by moving the index ind1 of s1 one step forward.
 - If ind2 is within bounds of s2 and the character at s2[ind2] matches the character at s3[ind1+ind2], it recursively calls solve by moving the index ind2 of s2 one step forward.

The ans is updated using the bitwise OR operation (|) to retain any previous true value and to combine the results of the two recursive calls.

5. Finally, the function stores the computed ans in the memoization table dp for the current ind1 and ind2 indices and returns this result.
6. The isInterleave function is the main function that's called to determine whether s3 can be formed by interleaving characters from s1 and s2. It first checks if the total length of s1 and s2 matches the length of s3. If not, it returns false as it's impossible to form s3.

7. It initializes a 2D vector `dp` to store the memoization table. The dimensions of this table are `(s1.size() + 1)` rows and `(s2.size() + 1)` columns, with all values initialized to -1.
8. It then calls the `solve` function with initial indices `ind1` and `ind2` set to 0, along with the memoization table `dp`. The result of this call indicates whether it's possible to form `s3` by interleaving `s1` and `s2`.
9. The `isInterleave` function returns the result obtained from the `solve` function.

Complexity

- Time complexity: $O(n*m)$
- Space complexity: $O(n*m)$