```
# Psuedocode:
'''
Interleaving of 2 strings s1 and s2 forms s3
if len of s3 is not equal to sum of s1 and s2, it is a wrong interleaving.
splitString() splits a string and forms all possible substrings
We have 3 more central menthods checkInterleaving(), dynamicProgramme(), backtracking()
checkInterleaving() passes s1, s2 and s3 to for forming a 3d Dynamic programming matrix
dynamicProgramming():
    form a 3d matrix "dp" with all values filled with False value, with x-axis as s1 subtsring, y-axis as s2 substring, z-axis as s3
    dp[0][0][0] = True
    for x in range(s1SplitLen+1):
        for y in range(s2SplitLen+1):
            for z in range(1, s3Len+1):
                # Checking the match of substring from s1 with s3
                if last portion of s3 == current s1 substring and x>0 & z has more space left than previous x & dp[previous x][y][previous z] == True:
                    dp[x][y][z] = True
                # Checking the match of substring from s2 with s3
                if last portion of s3 == current s2 substring and y>0 & z has more space left than previous y & dp[x][y-1][previous z] == True:
                    dp[x][y][z] = True
    return dp

backtracking():
    if dp[s1SplitLen][s2SplitLen][len(s3)]:
        count += 1
        while s1SplitLen > 0 or s2SplitLen > 0:
            # check for s1
            interleaving.append((s1Split[s1SplitLen - 1], "s1"))
            s3Len = s3Len - len(s1Split[s1SplitLen - 1])
            s1SplitLen = s1SplitLen - 1
            #similarly check for s2
            interleaving.append((s2Split[s2SplitLen - 1], "s2"))
            s3Len = s3Len - len(s2Split[s2SplitLen - 1])
            s2SplitLen = s2SplitLen - 1

'''
```

```
'''
Informal argument for its correctness
s1: aabcc
s2: dbbca
s3: aadbbcbcac

Interleaving exists: True, Count of interleavings: 36
s1 substrings: aa, bc, c
s2 substrings: db, bca
'''


'''
Time complexity is exponential because of generation of substrings
Space complexity is exponential because of generation of substrings
'''
```