HQ10 // Let X and Y be two strings of lengths m and n respectively.

a) Now let us assume that X does not appear in Y and $X[1 \dots (m-1)]$ appears in $Y[1 \dots (n-1)]$.
Again it is given that $X[m] = Y[n]$. So if we consider $X[m]$ along with $X[1 \dots (m-1)]$ then we get the entire X, and if we consider $Y[n]$ along with $Y[1 \dots (n-1)]$ then we get entire Y also, and considering all the indices of X and Y we can say that $X[1 \dots m]$ appears in $Y[1 \dots n]$, which basically implies X appears in Y. Therefore we got a contradiction to our initial assumption that X does not appear in Y.
Hence proved, that X appears in Y iff $X[1 \dots (m-1)]$ appears in $Y[1 \dots (n-1)]$.

b) Let us assume that X does not appear in Y and $X[1 \dots m]$ appears in $Y[1 \dots (n-1)]$. Again it is given that $X[m] \neq Y[n]$ and also from our assumption we get $X[m]$ appears in $Y[1 \dots (n-1)]$. Therefore considering all indices of X, we can conclude that X appears in $Y[1 \dots (n-1)]$ which contradicts our initial assumption.
Hence proved that, X appears in Y iff $X[1 \dots m]$ appears in $Y[1 \dots (n-1)]$.

c) Given i, j, k compute

1. $DS(i,j,k)$ = whether $X_1[0 \dots i]$ and $X_2[0 \dots j]$ appears in $Y[0 \dots k]$ or not

where $X_1 = X_2 = X$. It returns False if X does not appear twice in Y as disjoint subsequence and True if it appears at least twice as disjoint subsequence in Y.

<u>Note</u>: assuming that indexing of X_1, X_2 and Y starts from 0.

(2) $DS(i, j, k) = DS(i-1, j, k-1)$ AND $DS(i, j-1, k-1)$

OR $DS(i, j, k-1)$

$DS(0,0,k) = True$
$DS(i,j,0) = False$ } // base cases

(3) $X_1[m] = Y[n]$ then, X_1 appears in Y iff $X[0...m-1]$ appears in $Y[0...n-1]$.
also if $X_1[m] != Y[n]$ then X_1 appears in Y iff $X_1[0...m]$ appears in $Y[0...n-1]$. The same can be written for X_2. The equality condition ($X_1[m] = Y[n]$) will be taken care by the recursive steps starting from the base case.
In our recurrence we also check the same thing i.e., ~~DS[i-1][j][k-1]~~ $DS(i-1, j, k-1)$ checks if $X_1[0... i-1]$ appears in $Y[0...k-1]$ and $DS(i, j-1, k-1)$ check if $X_2[0...j-1]$ present in $Y[0...k-1]$ as subsequence.
We do a AND because we want both X_1 and X_2 to be present in Y. The last recurrence handles the case where $X_1[i] \neq Y[k]$ or $X_2[j] \neq Y[k]$. We are doing a OR in this part as there must be characters in Y which will not match with X_1 and X_2 but which should not stop our recurrence or make False return. With these 3 cases we will calculate if $X_1[0...i]$ and $X_2[0...j]$ appears in $Y[0...k]$.

If there is no such X_1 and X_2 i.e., no such X empty string then we can simply return True, i.e, X appears at least twice in Y, as empty string is always a subsequence of whole string.
While if Y is empty but X is not empty then we return False, i.e, X does not appear atleast twice in Y.

4 Let $S[0 \ldots m][0 \ldots m][0 \ldots n]$ be a 3D array.

$S[i][j][k]$ will store the value of $DS(i, j, k)$.

5 Initialize $S[0][0][k] = \text{True} \quad \forall k \in [0 \ldots n]$

$S[i][j][0] = \text{false} \quad \forall i, j \in [0 \ldots m]$

```
for i = 1 to m
    for j = 1 to m
        for k = 1 to n
            compute S[i][j][k] = DS(i,j,k)
    // using the recursive formula on sequence X1, X2, Y.
```

6 Whether X appears at least twice in Y as disjoint subsequence $= DS(m, m, n)$

7 Space Complexity $= O(m^2 n)$

Time Complexity $= O(m^2 n)$

each computation of $DS(i, j, k)$ requires $O(1)$ time and there $O(m^2 n)$ computation for X_1, X_2 with Y.