1: For bit strings $X = x_1 \dots x_m$, $Y = y_1 \dots y_n$ and $Z = z_1 \dots z_{m+n}$, we say that Z is an interleaving of X and Y if it can be obtained by interleaving the bits in X and Y in a way that maintains the left-to-right order of the bits in X and Y . For example, if X = 1101 and Y = 01 then $x_1x_2y_1x_3y_2x_4 =$ 110011 is an interleaving of X and Y . Give the most efficient algorithm you can to determine if Z is an interleaving of X and Y. (10)

Solution :

Optimal Substructure

Define $z_1 \dots z_{i+j}$ is an interleaving of $X = x_1 \dots x_i$ and $Y = y_1 \dots y_j$ for $0 \leq i \leq m$ and $0 \leq j \leq n$ Let $c[i, j]$ be true if and only if $z_1 \dots z_{i+j}$ is an interleaving of $X = x_1 \dots x_i$ and $Y = y_1 \dots y_j$. Now, the subproblem $c[i, j]$ can be defined recursively as below:

$$c[i,j] = \begin{cases} true & \text{if } i = j = 0 \\ false & \text{if } x_i \neq z_{i+j} \wedge y_j \neq z_{i+j} \\ c[i-1,j] & \text{if } x_i = z_{i+j} \ \wedge y_j \neq z_{i+j} \\ c[i,j-1] & \text{if } x_i \neq z_{i+j} \wedge y_j = z_{i+j} \\ c[i-1,j] \vee c[i,j-1] & \text{if } x_i = z_{i+j} \wedge y_j = z_{i+j} \end{cases}$$

$c[m, n]$ gives the answer to the original problem.

Correctness Proof :
First case: when $i = j = 0$, both X and Y are empty and then by definition Z is also empty and therefore, Z is a valid interleaving of X and Y.
Second case: if $x_i \neq z_{i+j}$ and $y_j \neq z_{i+j}$ then there is no interleaving possible and therefore it returns false.
Third case: if $x_i = z_{i+j}$ and $y_j \neq z_{i+j}$ then there exist a valid interleaving of X_i and Y_j in which x_i appears last, if and only if, a valid interleaving exists for X_{i-1} and Y_j.
Fourth case: This is symmetric to the third case.
Fifth case: When both Third, and Fourth cases are satisfied, we can find a valid interleaving of X and Y, by extending either through the Third case or through the Fourth case.

Clearly the optimal substructure property holds, because a correct interleaving of X and Y must contain in it correct interleaving of a subproblem involving X_i and Y_j, for $0 \leq i \leq m$, and $0 \leq j \leq n$.

Pseudo code:

String-Interleaving($X[1..m], Y[1..n], Z[1..p]$)

```
1  c[0..m, 0..n] is a 2-dimensional array (default entry: false)
2  if m = n = p = 0
3      return true
4  if p ≠ m + n
5      return false
6  if m == 0 and Y == Z
7      return true
8  else
9      return false
10 if n == 0 and X == Z
11     return true
12 else
13     return false
14 c[0,0] = true;
15 for i = 1 to m
16     for j = 1 to n
17         c[i,j] = (X[i] == Z[i+j] ∧ c[i−1,j]) ∨ (Y[j] == Z[i+j] ∧ c[i,j−1])
18 return c[m,n]
```

Running Time :
There are $m \times n$ cells in the memoization table, and filling each cell takes constant time using Line 17. Therefore the time complexity is $O(mn)$