

```

int stringMatching(ITEM[] P, ITEM[] T, int m, int n)

    // inizializzo i casi base
    int[][] DP ← new int[0...m][0...n]
    da j ← 0 fino a n fai                                     // Caso base: j ← 0
    |   DP[0][j] ← 0
    da i ← 1 fino a m fai                                     // Caso base: i ← 0
    |   DP[i][0] ← i

    // riempio la tabella
    da i ← 1 fino a m fai                                     // Caso generale
    |   da j ← 1 fino a n fai
    |   |   int temp ← DP[i - 1][j - 1] + if(P[i] ← T[j], 0, 1)           // 0: uguali, 1:
    |   |   |   sostituzione
    |   |   |   temp ← min(temp, DP[i - 1][j] + 1)                       // DP[i - 1][j] inserimento
    |   |   |   temp ← min(temp, DP[i][j - 1] + 1)                       // DP[i][j - 1] rimozione
    |   |   |   DP[i][j] ← temp
    |   |
    |   // cerco la posizione del minimo
    |   int min ← DP[m][0] // minimo
    |   int pos ← 0 // posizione del minimo
    |   da j ← 1 fino a n fai // trova il minimo sull'ultima riga
    |   |   se DP[m][j] < min allora
    |   |   |   min ← DP[m][j]
    |   |   |   pos ← j

    ritorna pos // potrei anche restituire DP[m][pos] che è il valore min associato

```