

```

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

    // inizializzo i casi base
    int[ ][ ] DP ← new int[0...m][0...n]
    from j ← 0 until n do                                     // Caso base: j ← 0
    |   DP[0][j] ← 0

    from i ← 1 until m do                                       // Caso base: i ← 0
    |   DP[i][0] ← i

    // riempio la tabella
    from i ← 1 until m do                                       // Caso generale
    |   from j ← 1 until n do
    |   |   int temp ← DP[i - 1][j - 1] + iif(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
    |   from j ← 1 until n do // trova il minimo sull'ultima riga
    |   |   if DP[m][j] < min then
    |   |   |   min ← DP[m][j]
    |   |   |   Pos ← j
    |
    return Pos // potrei anche restituire DP[m][Pos] che è il valore min associato

```