

8'5

## Problema 48

GONZÁLEZ RAMIRO, IVÁN (TAIS50)

ID envío	Usuario/a	Hora envío	Veredicto
63658	TAIS50	2022-11-23 10:29	AC
63599	TAIS50	2022-11-23 09:56	AC

Fichero Source.cpp

\*  
 \* Indicad el nombre completo y usuario del juez de quienes habéis hecho esta solución:  
 \* Estudiante 1: MICHAEL PAREDES SANCHEZ TAIS74  
 \* Estudiante 2: IVÁN GONZÁLEZ RAMIRO TAIS50  
 \*

Programacion dinamica, del tipo descendente.

Se necesita una matriz de n filas y m columnas (siendo n el tamaño de la primera palabra y m de la segunda)

para reconstruir la solución.

Llamada inicial

subsecuencia(0,0)

La recurrencia es la siguiente

subsecuencia(i+1,j+1)+1 si palabra1[i]==palabra[j]

max(subsecuencia(i+1,j), subsecuencia(i,j+1)) si palabra1[i] != palabra2[j]

CASOS BASE

subsecuencia(palabra1.size(),j)=0

subsecuencia(i,palabra2.size())=0

n = Y.size()

```

0...i...X.size()
-----
. | 5 ----- 0 |
. |      *-----0 |
j |                  |
. |                  |
. |                  |
. |                  1 0 |
n |                  0 0 0 |
-----

```

```
int subsecuencia(string& X, string& Y, int i, int j, Matriz<int> &m) {
```

```
    int& res = m[i][j];
```

```
    if (res == -1) {
```

```
        if (i >= X.size() || j >= Y.size()) {
```

```
            res = 0;
```

```
        }
```

```
        else if (X[i] == Y[j]) res = subsecuencia(X, Y, i + 1, j + 1, m) + 1;
```

```
        else {
```

```

        res = max(subsecuencia(X, Y, i + 1, j, m), subsecuencia(X, Y, i, j + 1, m));
    }
}
return res;
}
string reconstruir(string const& X, string const& Y, int i, int j, Matriz<int> const& m) {
    if (i >= X.size() || j >= Y.size()) return {};
    if (X[i] == Y[j])
        return X[i] + reconstruir(X, Y, i + 1, j + 1, m);
    else if (m[i][j] == m[i + 1][j])
        return reconstruir(X, Y, i + 1, j, m);
    else
        return reconstruir(X, Y, i, j + 1, m);
}

bool resuelveCaso() {
    // leemos la entrada
    string X, Y;
    cin >> X >> Y;

    if (!cin)
        return false;
    Matriz <int> m(X.size()+1, Y.size()+1, -1);
    subsecuencia(X, Y, 0, 0, m);
    cout << reconstruir(X, Y, 0, 0, m) << "\n";
    return true;
}

```