Programación Dinámica

JOAQUÍN ARCILA PÉREZ
LAURA LÁZARO SORALUCE
CRISTÓBAL MERINO SÁEZ
ÁLVARO MOLINA ÁLVAREZ

ÍNDICE

INTRODUCCIÓN

PROGRAMACIÓN DINÁMICA

- Naturaleza n-etápica del problema
- Verificación del Principio de Optimalidad de Bellman
- Planteamiento de una recurrencia
- Cálculo de la solución

DESARROLLO

Eficiencia teórica

O(n*m)

```
void MatrizCalculos (string seq1, string seq2,
vector<vector<int>> & matriz) {
 for (int j=0; j<=seq2.size(); j++) {
  for (int i=0; i<=seq1.size(); i++) {
    if (j==0 || i==0)
     matriz[i][j] = 0;
    else if (seq1[i-1] == seq2[j-1])
     matriz[j][i] = matriz[j-1][i-1]+1;
    else
     matriz[j][i]=max(matriz[j-1][i], matriz[j][i-1]);
 } } }
```

DESARROLLO

Matriz de cálculos

		a	b	b	c	d	e	f	a	b	c	d	x	z	y	c	c	d
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
a	0	↖1	←1	←1	←1	←1	←1	←1	↖1	←1	←1	←1	←1	←1	←1	←1	←1	←1
b	0	↑1	↖2	↖2	←2	←2	←2	←2	←2	↖2	←2	←2	←2	←2	←2	←2	←2	←2
b	0	↑1	↖2	↖3	←3	←3	←3	←3	←3	↖3	←3	←3	←3	←3	←3	←3	←3	←3
c	0	↑1	↑2	↑3	↖4	←4	←4	←4	←4	←4	↖4	←4	←4	←4	←4	↖4	↖4	←4
d	0	↑1	↑2	↑3	↑4	↖5	←5	←5	←5	←5	←5	↖5	←5	←5	←5	←5	←5	↖5
e	0	↑1	↑2	↑3	↑4	↑5	↖6	←6	←6	←6	←6	←6	←6	←6	←6	←6	←6	←6
a	0	↖1	↑2	↑3	↑4	↑5	↑6	↑6	↖7	←7	←7	←7	←7	←7	←7	←7	←7	←7
f	0	↑1	↑2	↑3	↑4	↑5	↑6	↖7	↑7	↑7	↑7	↑7	↑7	↑7	↑7	↑7	↑7	↑7
b	0	↑1	↖2	↖3	↑4	↑5	↑6	↑7	↑7	↖8	←8	←8	←8	←8	←8	←8	←8	←8
c	0	↑1	↑2	↑3	↖4	↑5	↑6	↑7	↑7	↑8	↖9	←9	←9	←9	←9	↖9	↖9	←9
d	0	↑1	↑2	↑3	↑4	↖5	↑6	↑7	↑7	↑8	↑9	↖10	←10	←10	←10	←10	←10	↖10
z	0	↑1	↑2	↑3	↑4	↑5	↑6	↑7	↑7	↑8	↑9	↑10	↑10	↖11	←11	←11	←11	←11
x	0	↑1	↑2	↑3	↑4	↑5	↑6	↑7	↑7	↑8	↑9	↑10	↖11	↑11	↑11	↑11	↑11	↑11
y	0	↑1	↑2	↑3	↑4	↑5	↑6	↑7	↑7	↑8	↑9	↑10	↑11	↑11	↖12	←12	←12	←12
c	0	↑1	↑2	↑3	↖4	↑5	↑6	↑7	↑7	↑8	↖9	↑10	↑11	↑11	↑12	↖13	↖13	←13
c	0	↑1	↑2	↑3	↖4	↑5	↑6	↑7	↑7	↑8	↖9	↑10	↑11	↑11	↑12	↖13	↖14	←14
d	0	↑1	↑2	↑3	↑4	↖5	↑6	↑7	↑7	↑8	↑9	↖10	↑11	↑11	↑12	↑13	↑14	↖15

DESARROLLO
PRIMER EJEMPLO

Secuencias:

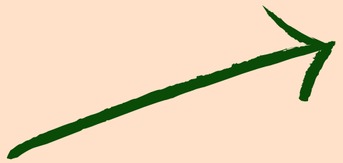
- abbcdefabcdxzyccd
- abbcdeafbcdzxyccd

DESARROLLO

Resultados primer ejemplo

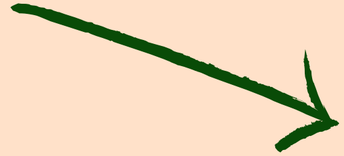
Longitud de la secuencia más larga: 15

Subsecuencia 1: abbcdefbcdxyccd



abbcdefabcdxzyccd
abbcdeafbcdzxyccd

Subsecuencia 2: abbcdeabcdzyccd



abbcdefabcdxzyccd
abbcdeafbcdzxyccd

Porcentaje de parecido: 88.2353%

DESARROLLO
SEGUNDO EJEMPLO

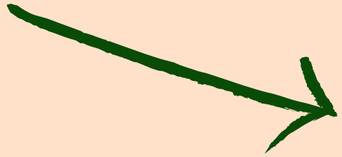
Secuencias:

- 0101110001000101010100100010010001001
- 1100001001001010100010100100110101100

DESARROLLO

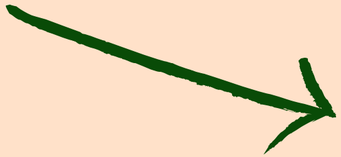
Resultados segundo ejemplo

Subsecuencia 1: 10000100010101000100010010010030)



0101110001000101010100100010010010011000010010010010101000101001001101010

Subsecuencia 2: 11000000101010100100010010010030)



0101110001000101010100100010010010011000010010010010100010100100110101000

Porcentaje de parecido: 83.3333%

CONCLUSIÓN

- La PD nos proporciona resultados óptimos, cosa que no siempre se cumple en el caso de otros algoritmos como los Greedy.

- Conseguimos implementar algoritmos más eficientes que por fuerza bruta.

MUCHAS GRACIAS