Programacion Dinamica

Definición

Problemas de optimisacion que se puede dividir en subproblemas independientes de tal manera que la solucion optima al problema se puede deducir de las soluciones optima a los subproblemas.

Ejemplos

- Fibonacci
- Torre de Hanoi (---)] recurrencias lineales → solucion "directa"

- Subsecuencia mas larga
- Multiplicacion de Cadenas de Matrices
- Cambio de Moneda
- Arboles de Busqueda Optimos

① Fibonacci

$$\begin{cases} F_0 = F_1 = 1 \\ F_{i+2} = F_i + F_{i+1} \end{cases}$$

Complejidad

Tiempo $\Theta(F_n) = O(1.5^n)$?

Espacio $\Theta(n)$

```
Fibonacci(n) =
    if n=0 o n=1 return 1
    else Return Fibonacci(n-1) + F(n-2)
```

Fiboracci Mas Listo Pero Todavia Tonto (n)!

⟶ con un arreglo $F[1..n]$

- set $F[0]=1$, $F[1]=1$,
- Para $i=2$ hasta n
  $F[i] \leftarrow F[i-2]+F[i-1]$;
- Return $F[n]$

Complejidad
Tiempo $O(n)$
Espacio $O(n)$

Con la Formula,

Tiempo ? = tiempo para calcular $(\sqrt{5}^n)$

Espacio ? ⟶ Multiplicacio Egipcia

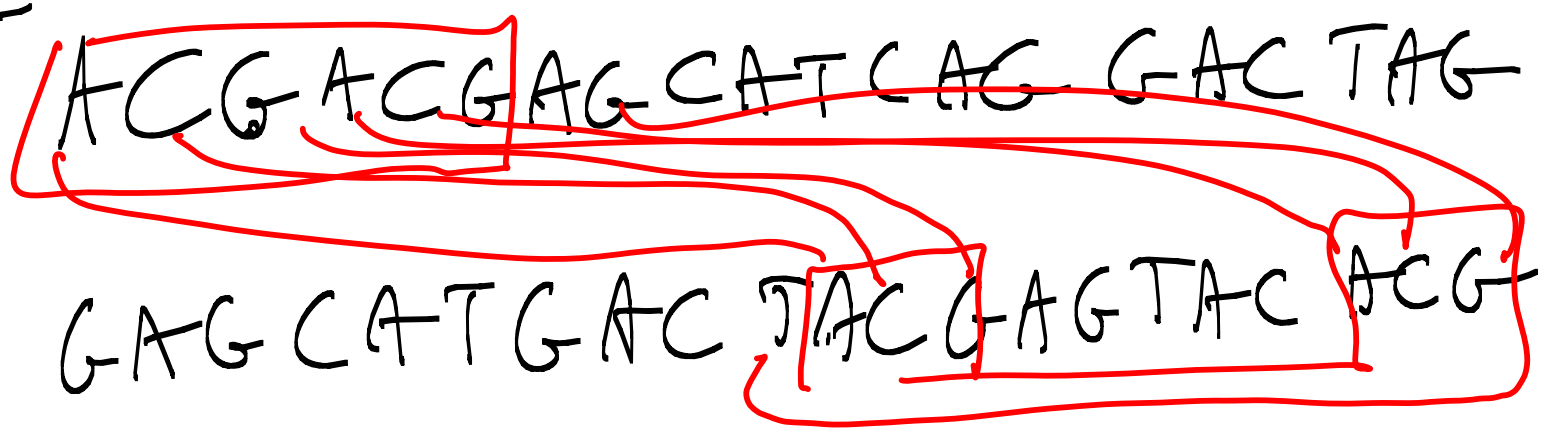
② Subsecuencia comun mas Larga

Definicion

Dado dos textos T_1, T_2
de tamaño n, m

Encontrar la subsecuencia comun mas larga

Ejemplo



Solucion

$$I_{i,j} = \begin{cases} 1 & \text{si } T_1[i] = T_2[j] \\ 0 & \text{sino} \end{cases}$$

$M[i,j]$ = tamaño de la subsecuencia comun mas larga entre $T_1[1,\dots,i]$ y $T_2[1\dots j]$

- Caso Inicial $M[1,1] = \begin{cases} 1 & \text{si } I_{1,1} = 1 \\ 0 & \text{sino} \end{cases}$

Recurrencia

$$M[i+1, j+1] = \begin{cases} M[i, j] + 1 & \text{si } T_1[i+1] = T_2[j+1] \\ \max \begin{cases} M[i+1, j] \\ M[i, j+1] \end{cases} & \text{sino.} \end{cases}$$

Complejidad

Tiempo : $\Theta(n \times m)$

Espacio : $\Theta(n + m)$ ~~$\Theta(n + m)$~~ $O(n+m)$

Multiplicacion de Secuencias de Matrices

$$M_1 \times M_2 \times M_3 \times \cdots \times M_k$$

$\longrightarrow$ a ver en Casa