

Antecedente: Divide y vencerás

1. Divides un problema en subproblemas.
2. Resuelves cada subproblema.
3. Combinas resultados.

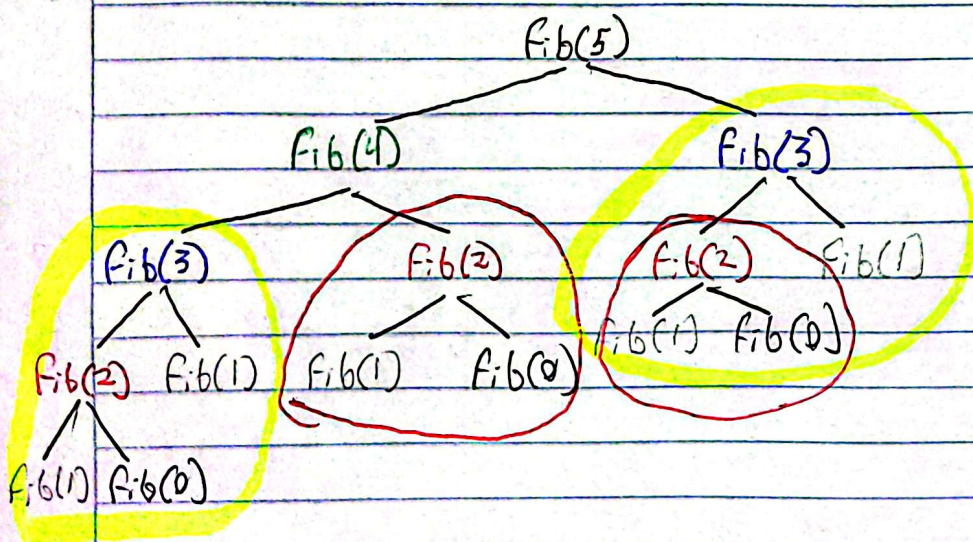
Programación dinámica

Sugiere la solución de cada subproblema una única vez y guarda un registro de los resultados de donde la solución al problema original pueda ser obtenida.

Caso: ~~Números~~ Números Fibonacci

Se pueden calcular mediante:-

$F(n) = F(n-1) + F(n-2)$ para cada $n > 1$, con condiciones iniciales $\begin{cases} F(0) = 0 \\ F(1) = 1 \end{cases}$



memoria

Solemos sacrificar ~~tiempo de ejecución~~ a cambio de tiempo de ejecución, pero bien optimizado se minimiza ese sacrificio.

Pasos para desarrollar un algoritmo de este tipo:

1. Describir la estructura de la solución óptima.

2- Definir de forma recursiva el valor de la solución óptima.

3- Calcular el valor óptimo usando programación dinámica (guardando valores ya calculados).

4- Construir solución óptima a partir de la información ya calculada (opcional).