

Programación Dinámica

Análisis y Diseño de Algoritmos

Dr. Jaime Osorio Ubaldo

La programación dinámica permite mejorar el rendimiento de algunos algoritmos. En ciertos casos, cuando se divide un problema en varios subproblemas, resulta ser que éstos no son independientes entre sí, es decir, cuando se resuelve un subproblema traslapado con otros, esto genera que se tenga que procesar nuevamente la solución generando un mayor costo temporal. En algunos casos puede generar un algoritmo de complejidad exponencial. La programación dinámica consiste en conservar la solución de cada subproblema que ya se ha resuelto en una tabla (puede ser un arreglo o una matriz), para tomarla cuando ésta se requiera. Esto reduce el tiempo necesario para obtener el resultado final, pues evita repetir algunos de los cálculos.

“En la programación dinámica normalmente se empieza por los subcasos más pequeños, y por tanto más sencillos. Combinando sus soluciones, obtenemos las respuestas para subcasos de tamaños cada vez mayores, hasta que finalmente llegamos a la solución del caso original” [Brassard Bratley, 2008].

- 1 Plantear un algoritmo de solución recursivo.
- 2 Organizar las soluciones parciales en una tabla.
- 3 Modificar el algoritmo recursivo de manera que antes de hacer el llamado recursivo se consulte la tabla y, en caso de hallarse la solución en ésta, ya no se hace el llamado recursivo.

Aplicación: Sucesión de Fibonacci

La sucesión de Fibonacci es

$$1, 1, 2, 3, 5, 8, \dots$$

Recursivamente está definida por

$$F(n) = \begin{cases} 1, & n = 0, 1 \\ F(n-1) + F(n-2), & n \geq 2 \end{cases}$$

Algoritmo usando Programación Dinámica

```
Tabla[1...n] = 0
Fibonacci(n)
  Si  $n \leq 1$ 
    return 1
  Si Tabla[n] = 0
    Tabla[n] = Fibonacci(n - 1) + Fibonacci(n - 2)
  retorna Tabla[n]
```

Si bien en este algoritmo hay recursión, ésta solo se lleva a cabo cuando no hay datos en la tabla. Cada vez que se requiere un nuevo dato en la tabla se hace la recursión, sin embargo ésta no es muy profunda, ya que toma los datos que ya se encuentran calculados en la tabla sin necesidad de llegar cada vez al caso base.

- 1 En la programación dinámica va guardando la historia y construyendo la solución a partir de las soluciones óptimas de problemas más pequeños resueltos previamente.
- 2 En la programación dinámica el problema debe poder dividirse en varias etapas, y la decisión en una etapa se toma después de haber considerado los resultados de otras etapas más simples.

Temas de investigación.

- 1 ¿Qué ventajas y desventajas presenta esta técnica de diseño?
- 2 Explique una aplicación de esta técnica de diseño.