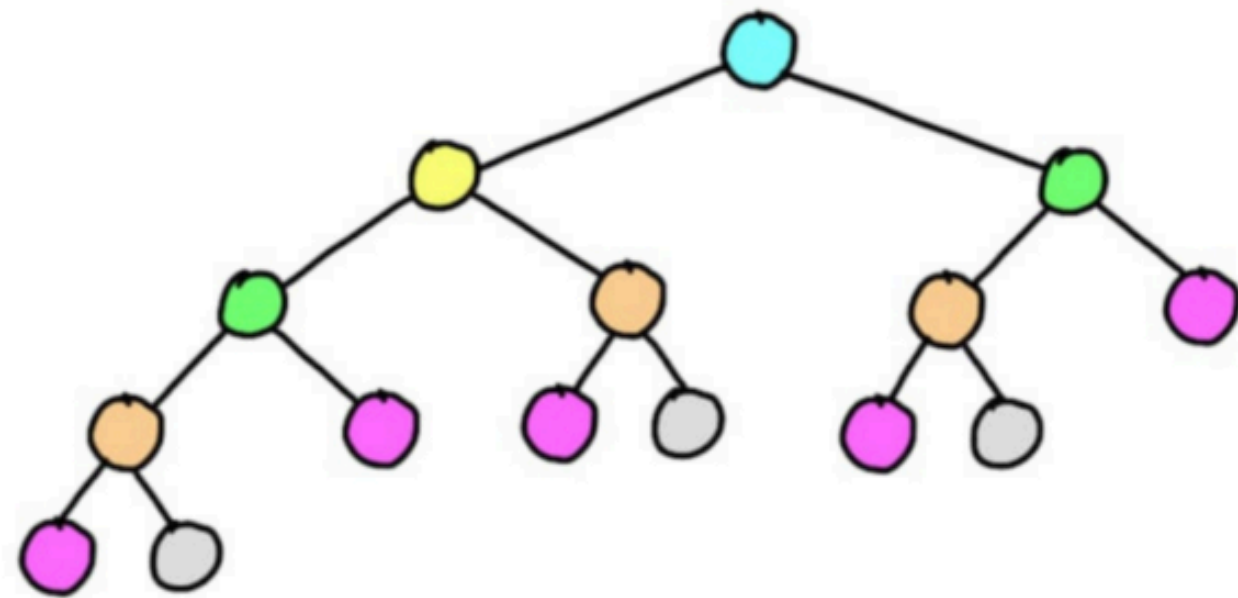


DYNAMIC PROGRAMMING

An Avid Coder

Always remember the past



Programación Dinámica (DP)

Programa Avanzado ICPC 2025

Por Waldo Caballero





Introducción a DP

Evitar Calculos Repetidos

- Usualmente, los problemas de programación dinámica se resuelven naturalmente mediante recursión. En estos casos, es más fácil escribir la solución recursiva y luego guardar los estados repetidos en una tabla de consulta. Este proceso se conoce como programación dinámica TOP DOWN con memoización.

Fibonacci

UNO DE LOS EJEMPLOS MÁS BÁSICOS Y CLÁSICOS DE ESTE PROCESO ES LA SECUENCIA DE FIBONACCI. SU FORMULACIÓN RECURSIVA ES:

$f(n) = f(n - 1) + f(n - 2)$ where $n \geq 2$ and $f(0) = 0$ and $f(1) = 1$. In C++, this would be expressed as:

```
int f(int n) {  
    if (n == 0) return 0;  
    if (n == 1) return 1;  
    return f(n - 1) + f(n - 2);  
}
```

EL TIEMPO DE EJECUCIÓN DE ESTA FUNCIÓN RECURSIVA ES EXPONENCIAL: APROXIMADAMENTE $O(2^N)$
YA QUE UNA LLAMADA DE FUNCIÓN
($F(N)$) DA COMO RESULTADO 2 LLAMADAS DE FUNCIÓN DE TAMAÑO SIMILAR
($F(N-1)$ Y $F(N-2)$).

Mejorando Fibonacci con DP (Memoizacion)

```
const int MAXN = 100;
bool found[MAXN];
int memo[MAXN];

int f(int n) {
    if (found[n]) return memo[n];
    if (n == 0) return 0;
    if (n == 1) return 1;

    found[n] = true;
    return memo[n] = f(n - 1) + f(n - 2);
}
```

EVITAMOS REPETIR "SUBPROBLEMAS"

Version Iterativa o Bottom Up

```
F[0] = 0  
F[1] = 1  
for i in 2 .. N:  
    F[i] = F[i-1] + F[i-2]
```

EN ESTE CASO FUE SENCILLO, PORQUE TENIAMOS LA FORMULA
PERO EN OTROS PROBLEMAS NO SERA TAN SENCILLO

Ways to Climb Stairs

You are climbing a staircase. It takes N steps to reach to the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?



¡ Gracias !

No dudes en preguntar o
escribirnos si tienes
alguna duda o consulta.