Programación Dinámica

* Almacenando valores
* Función: f

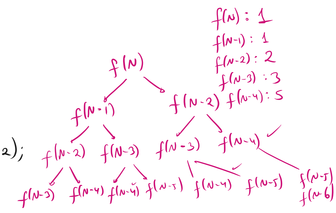
Ejm: Sea F_K el k-ésimo número de fibonacci. Hallar F_K, $1 \le K \le 10^6$

$F_0 = 0$
$F_1 = 1$
$F_n = F_{n-1} + F_{n-2}$, $n \ge 2$

Enfoque 1: Función recursiva

```
int fibo(int n){
  if(n ≤ 1) return n;
  return fibo(n-1) + fibo(n-2);
}
```

$O(2^n)$



$f(N-K) = F_{K+1} \Rightarrow f(1) : F_n \approx O(2^n)$

Enfoque 2: Prog Dinámica

f(N): N-ésimo número de fibonacci

```
vector<int> f(N+1);
f[0] = 0
f[1] = 1
for(int i=2; i≤N; i++){
  f[i] = f[i-1] + f[i-2]
}
```

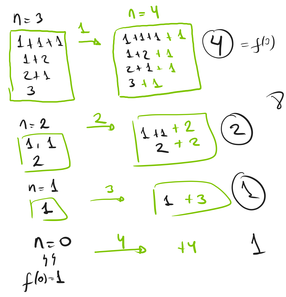
MOD = 10^9+7 → primos
MOD = 998244353

$f(i) \,\%\, MOD$

Dice Combinations

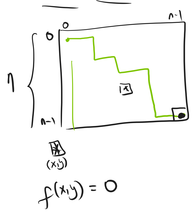
f(n): La cantidad de formas de lanzar dados obteniendo suma n

	último dado	suma n
f(n-1)	1	f(n)
f(n-2)	2	f(n)
f(n-3)	3	f(n)
f(n-4)	4	f(n)
f(n-5)	5	f(n)
f(n-6)	6	f(n)



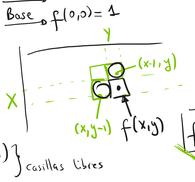
$f(n) = f(n-1) + f(n-2) + f(n-3) + f(n-4) + f(n-5) + f(n-6)$, $n \ge 1$
$f(0) = 1$
$f(K) = 0$, $K < 0$

Grid Paths



f(x,y): Total de caminos para llegar a la casilla (x,y)

Rec
Base → f(0,0) = 1



$f(x,y) = f(x-1,y)$
$f(x,y) = f(x,y-1)$

$f(x,y) = f(x-1,y) + f(x,y-1)$

$f(x,y) = 0$

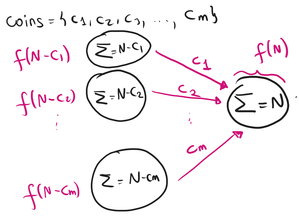
$f(x,y) = f(x-1,y) + f(x,y-1)$ } casillas libres
$f(0,0) = 0$
$f(x,y) = 0$ } casillas con trampa

Minimizing Coins

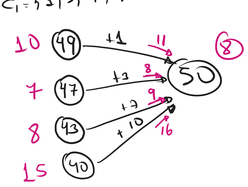
Suma deseada = K
mín cantidad de monedas para obtener suma = K

f(N): Mínima cantidad de monedas para obtener suma N

Coins = {$c_1, c_2, c_3, \dots, c_m$}



$c_1 = \{1, 3, 7, 10\}$



$f(N) = f(N-c_1) + 1$
$= f(N-c_2) + 1$
$f(N-cm) + 1$

$\Rightarrow f(N) = \min(f(N-c_1), f(N-c_2), \dots f(N-c_m)) + 1$

$f(0) = 0$
$f(N) = \infty$ si no es posible