Programacion Dinamica

Charles

Training Camp

Medellin 2026

Memoizacion

Exponencial

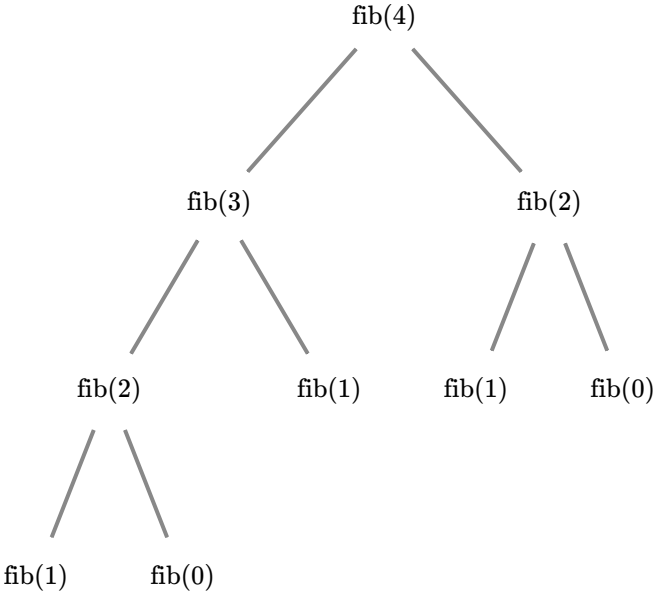
```
def fib(n):
    if n < 2: return n
    return fib(n-1) + fib(n-2)
```

Memoizado

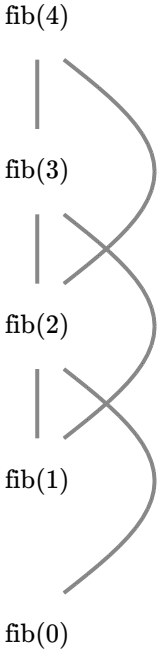
```
memo = {}
def fib(n):
    if n < 2: return n
    if n not in memo:
        memo[n] = fib(n-1) + fib(n-2)
    return memo[n]
```

Arbol vs DAG

Sin memo



Memoizado



Camino Minimo

4	7	2	9
6	1	8	3
5	9	4	2
3	2	6	1

Camino Minimo

$$dp_{i,j} = A_{i,j} + \min\left(dp_{i-1,j}, dp_{i,j-1}\right)$$

$dp_{i,j}$ es el costo minimo de un camino de $(0,0)$ a (i,j)

4	7	2	9
6	1	8	3
5	9	4	2
3	2	6	1

Patron muy comun

Una solucion es una serie de elecciones

Se quiere

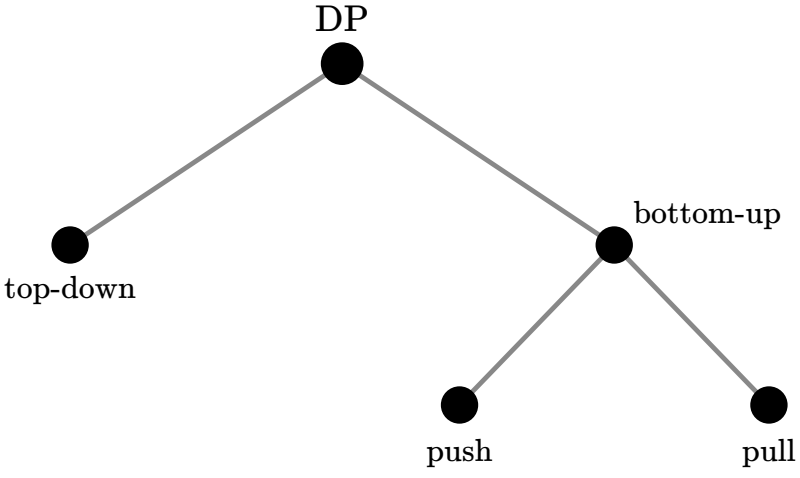
- La sol de menor costo
- La cantidad de sols
- Ver si existe una sol
- La sol minima lexicografica

Un prefijo de elecciones se puede comprimir

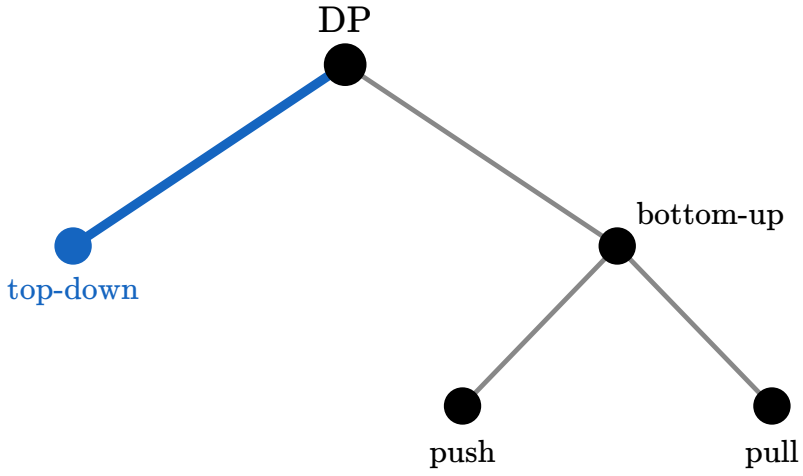
Alternaciones Comunes

Top-down y Bottom-up

Top-down / Bottom-up

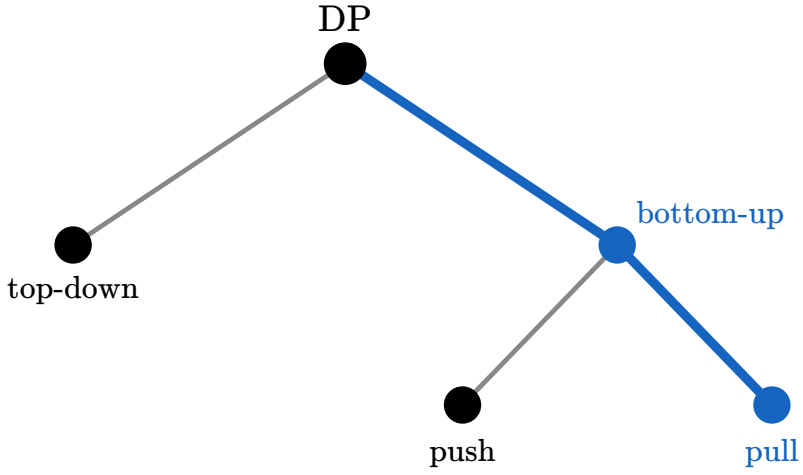


Top-down



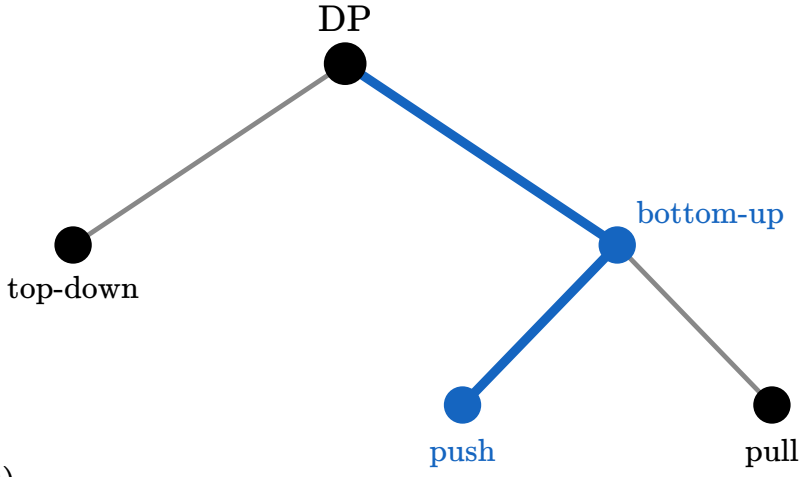

```
memo = {}
def fib(n):
    if n < 2: return n
    if n not in memo:
        memo[n] = fib(n-1) + fib(n-2)
    return memo[n]
```

Bottom-up (pull)




```
dp = [0, 1]
for i in range(2, n+1):
    dp.append(dp[i-1] + dp[i-2])
```

Bottom-up (push)




```
dp = [0] * (n+2)
dp[1] = 1
for i in range(n):
    dp[i+1] += dp[i]
    dp[i+2] += dp[i]
```

Pa’ lante y pa’ tra

Pa' lante y pa' tra

Pa' lante

mejor sufijo que parte desde el estado

$$dp_{i,j} = A_{i,j} + \min\left(dp_{i+1,j}, dp_{i,j+1}\right)$$

$dp_{i,j}$ = costo minimo de un camino $(i,j) \to (n,m)$

chico $\to$ grande

4	7	2	9
6	1	8	3
5	9	4	2
3	2	6	1

Pa' tra

mejor prefijo que llega al estado

$$dp_{i,j} = A_{i,j} + \min\left(dp_{i-1,j}, dp_{i,j-1}\right)$$

$dp_{i,j}$ = costo minimo de un camino $(0,0) \to (i,j)$

grande $\to$ chico

4	7	2	9
6	1	8	3
5	9	4	2
3	2	6	1

Compactar y Descompactar Coordenadas

Descompactar Coordenadas

$$dp_{i,j,c} = dp_{i-1,j,c-A_{i,j}} \text{ or } dp_{i,j-1,c-A_{i,j}}$$

$dp_{i,j,c}$ es verdadero si hay un camino de $(0,0)$ a (i,j) de costo exacto c

Knapsack

n items, cada uno con costo C_i y beneficio B_i

elegir un subconjunto con costo total $\leq K$

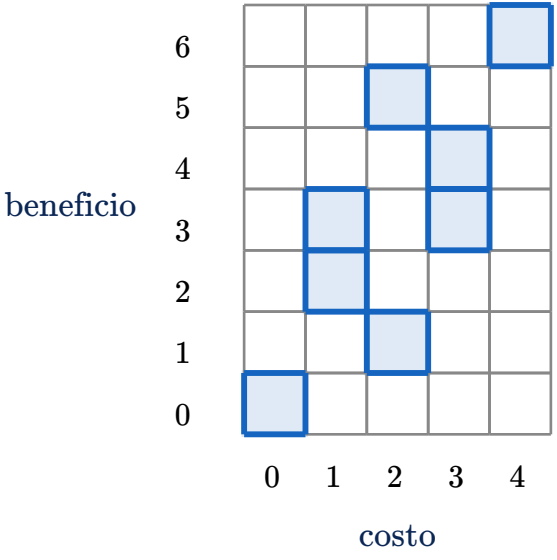
que maximice el beneficio total

Knapsack

$$dp_{i,c,b} = dp_{i-1,c,b} \text{ or } dp_{i-1,c-C_i,b-B_i}$$

$dp_{i,c,b}$ = existe subconjunto de los primeros i items

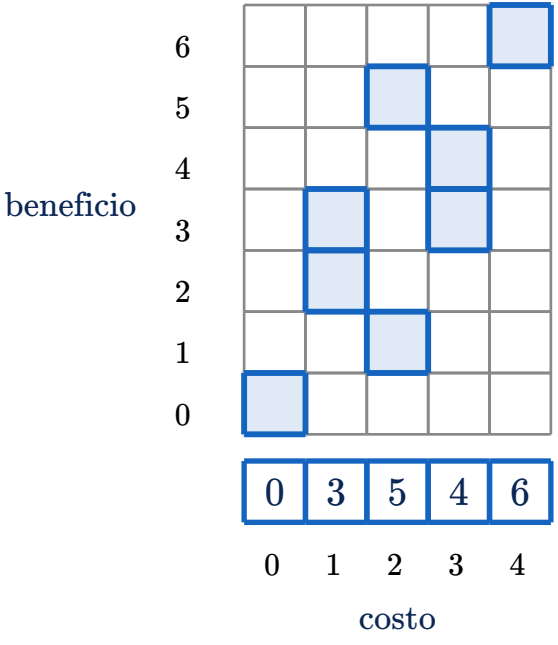
con costo c y beneficio b



Compactar Coordenadas

$$dp_{i,c} = \max\left(dp_{i-1,c}, dp_{i-1,c-C_i} + B_i\right)$$

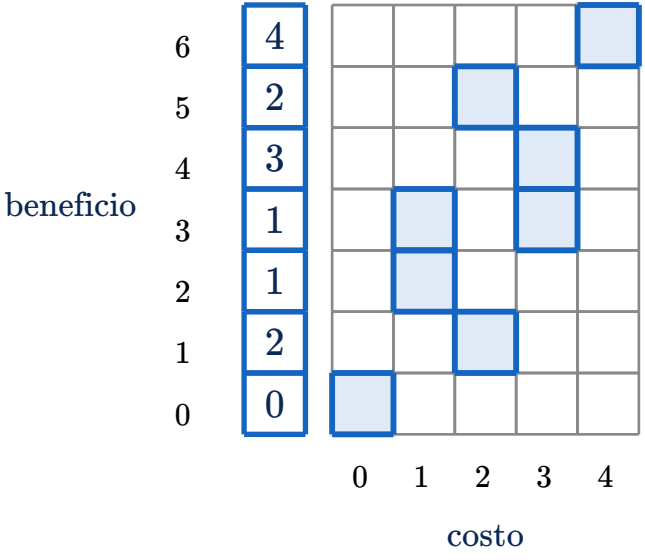
$dp_{i,c}$ = maximo beneficio con costo c usando i items



Compactar Coordenadas

$$dp_{i,b} = \min\left(dp_{i-1,b}, dp_{i-1,b-B_i} + C_i\right)$$

$dp_{i,b}$ = minimo costo con beneficio b usando i items

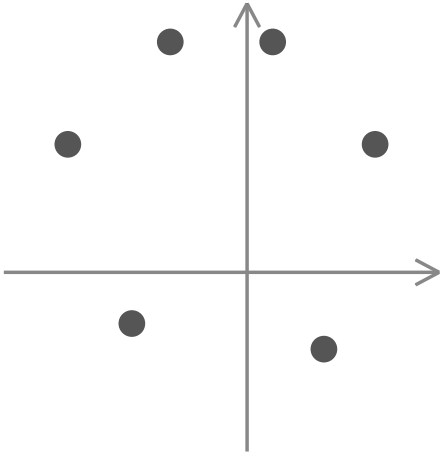


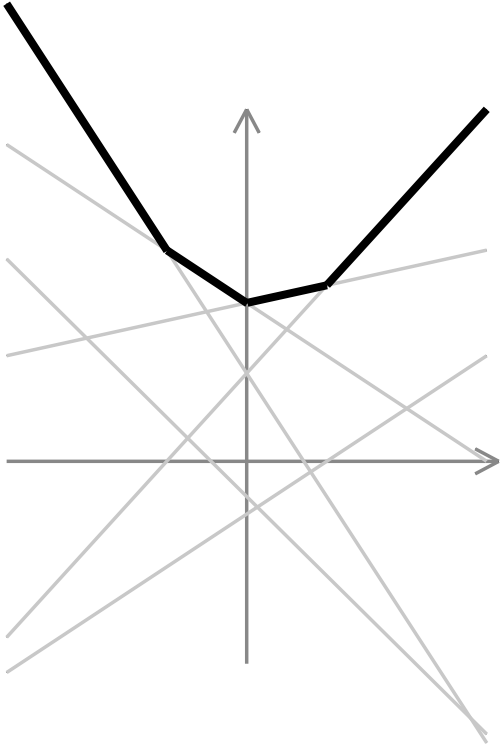
Break

10 minutos

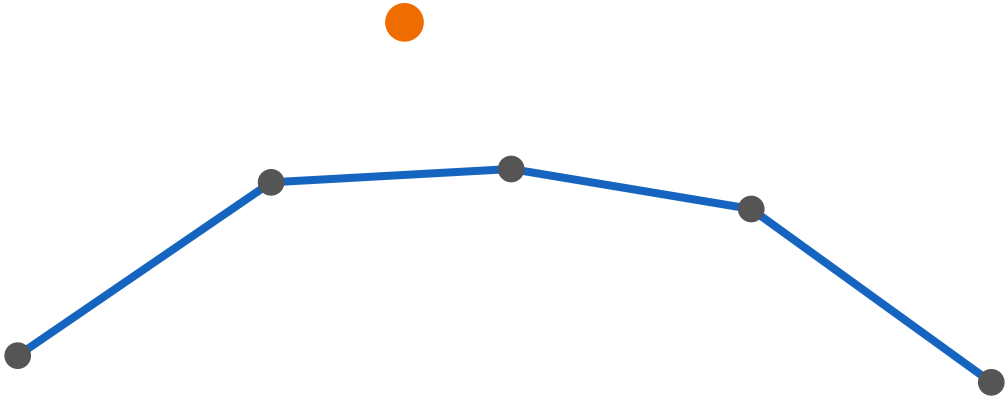
Convex Hull Trick

Envolvente Superior

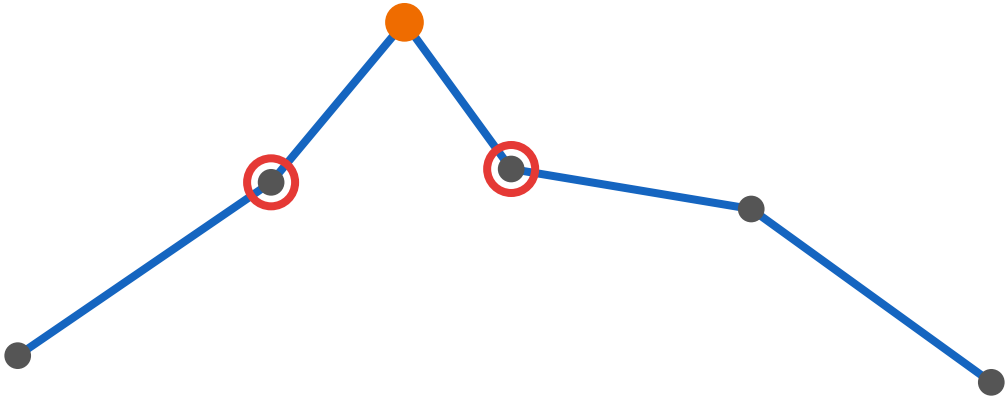




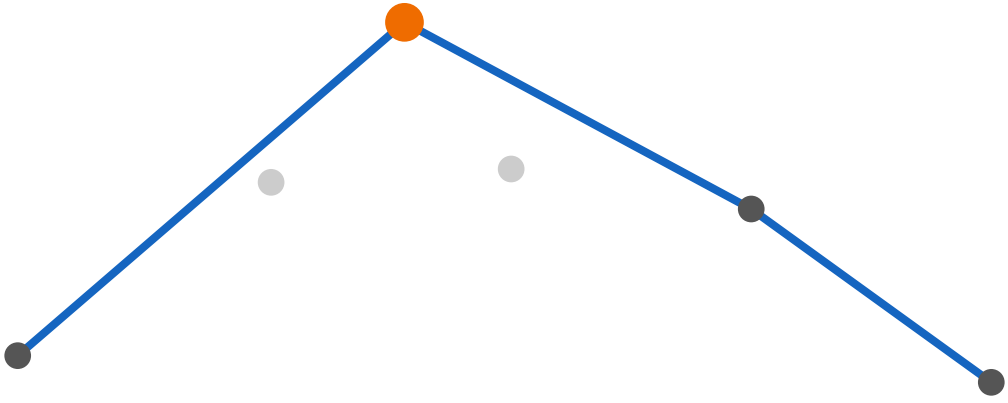
Insercion en el Medio



Insercion en el Medio



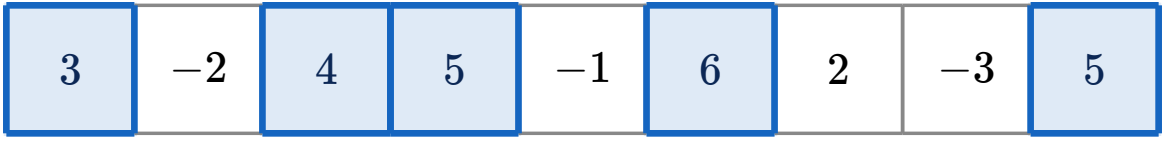
Insercion en el Medio



Ejemplo

elegir una subsecuencia $s_1, \ldots, s_k$ de A que maximice

$$s_1 s_2 + s_2 s_3 + \cdots + s_{k-1} s_k$$



Alien Trick

Alien Trick

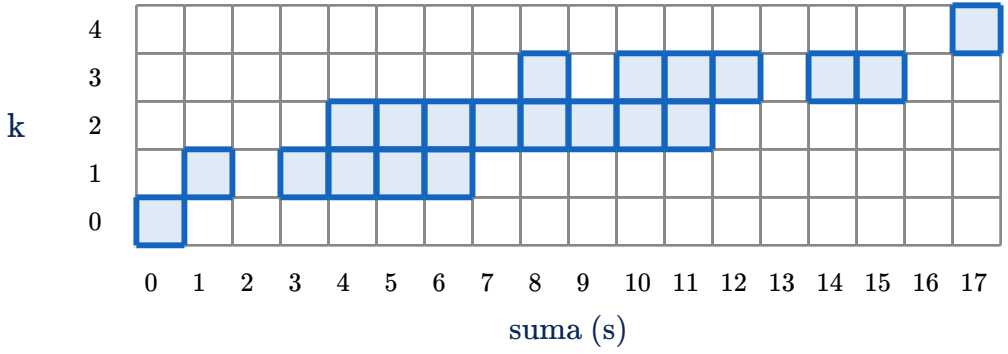
elegir K elementos de un array A, ninguno consecutivo,

que maximicen la suma

DP sin Compactar

$dp_{i,k,s}$ = se puede lograr suma s con k indices entre los primeros i

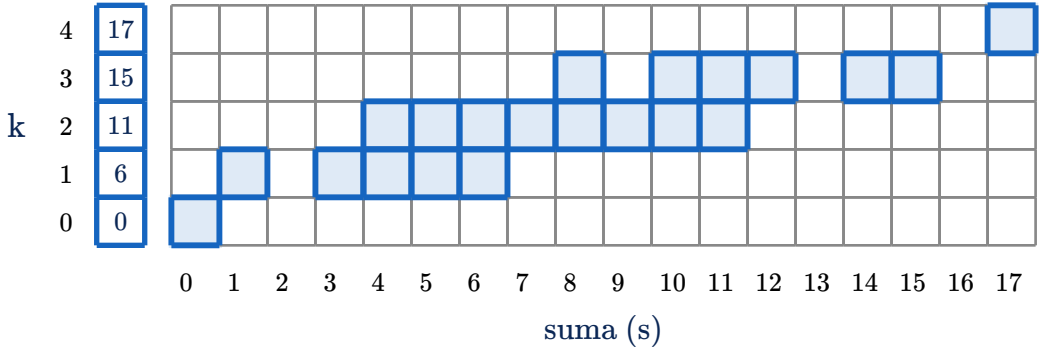
$$dp_{i,k,s} = dp_{i-1,k,s} \text{ or } dp_{i-2,k-1,s-A_i}$$



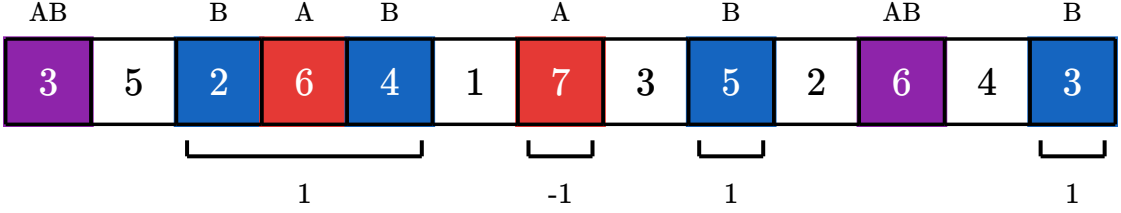
Compactar Coordenadas

$dp_{i,k}$ = maxima suma usando exactamente k indices entre los primeros i

$$dp_{i,k} = \max\left(dp_{i-1,k}, dp_{i-2,k-1} + A_i\right)$$



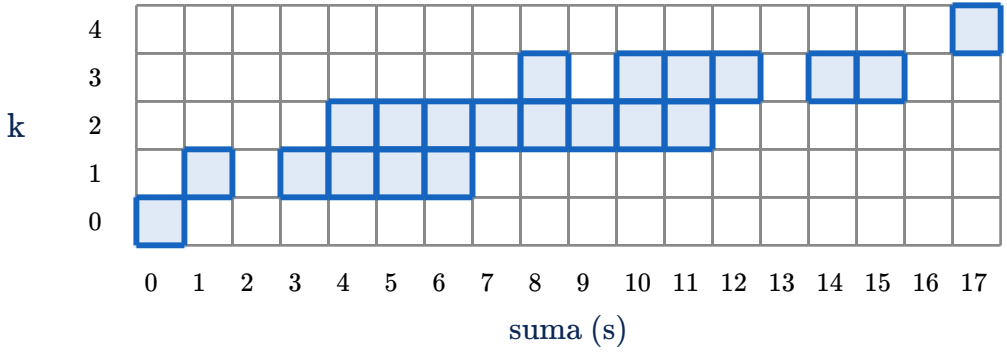
Concavidad



Alien Trick

dp_i^λ = maxima suma menos de elegir hasta i menos el costo

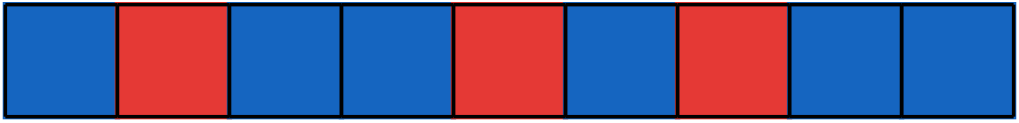
$$dp_i^\lambda = \max\left(dp_{i-1}^\lambda, dp_{i-2}^\lambda + A_i - \lambda\right)$$



Bonus: Front DP

Front DP

Cuantas formas hay de colorear un tablero de $1 \times n$ de azul y rojo tal que no haya celdas rojas adyacentes?

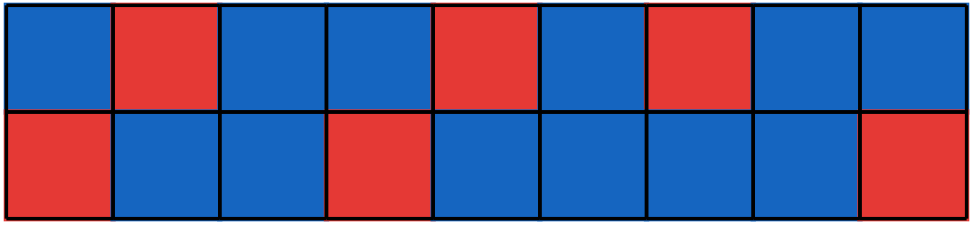


Front DP

Cuantas formas hay de colorear un tablero de $1 \times n$ de azul y rojo tal que no haya celdas rojas adyacentes?

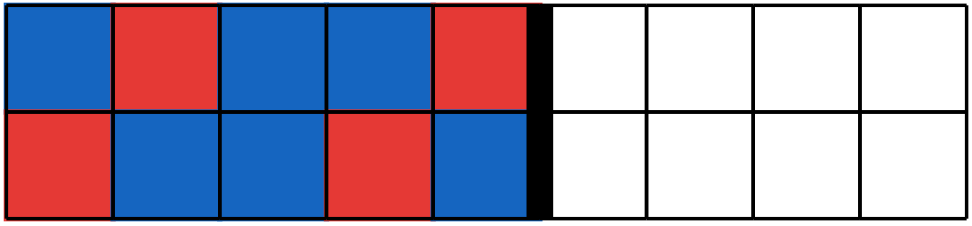
Front DP

Cuantas formas hay de colorear un tablero de $2 \times n$ de azul y rojo tal que no haya celdas rojas adyacentes?



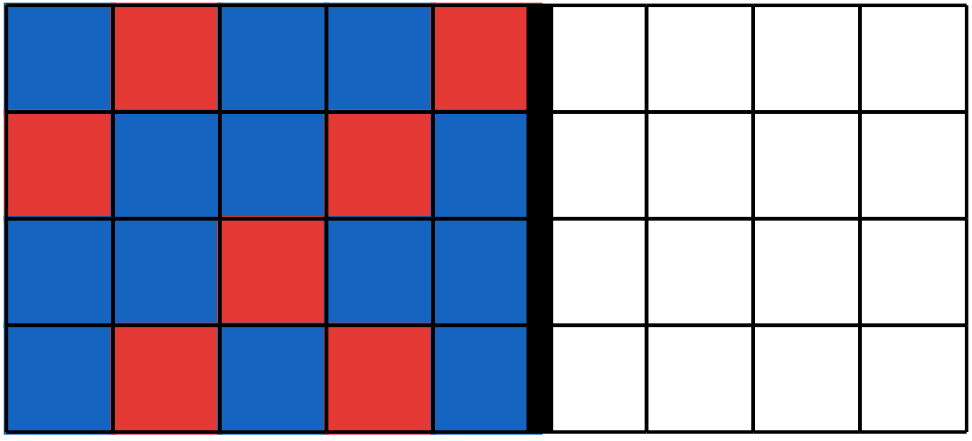
Front DP

Cuantas formas hay de colorear un tablero de $2 \times n$ de azul y rojo tal que no haya celdas rojas adyacentes?



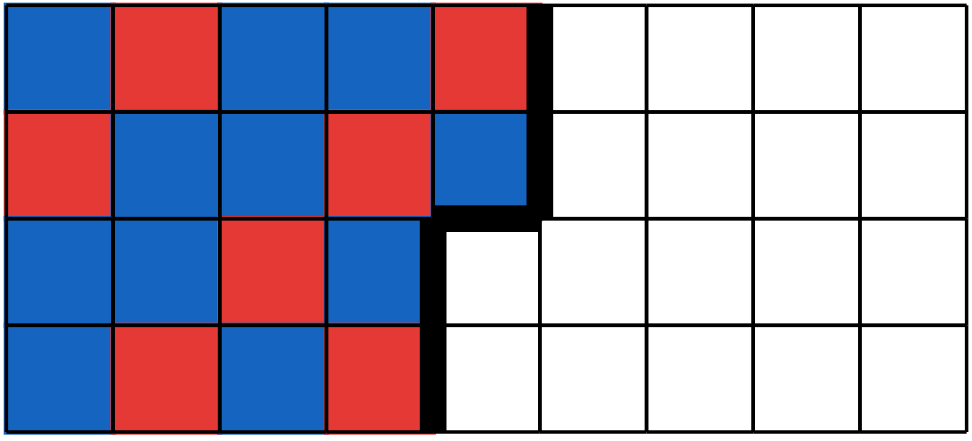
Front DP

Cuantas formas hay de colorear un tablero de $k \times n$ de azul y rojo tal que no haya celdas rojas adyacentes?



Front DP

Cuantas formas hay de colorear un tablero de $k \times n$ de azul y rojo tal que no haya celdas rojas adyacentes?



Fin!

reedef.dev/feedback