

Динамическое программирование

Фиб. число.

$$F_1 = 1 \quad F_2 = 1 \quad F_i = F_{i-1} + F_{i-2}$$

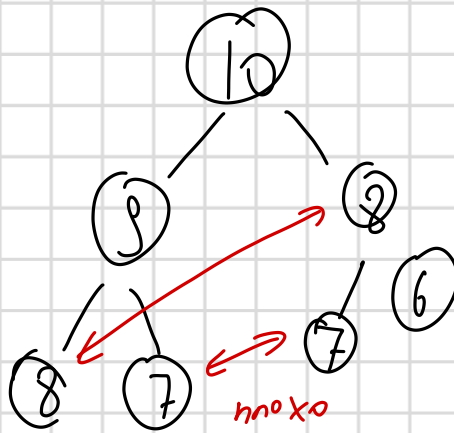
$f(i)$

if $(i \leq 2)$:

return 1

else: return $f(i-1) + f(i-2)$

$$T(n) = 1 + T(n-1) + T(n-2)$$



2-ой вариант

res = null

$f(i)$: $O(n)$

if $res[i] \neq null$: return res

else:

if $(i \leq 2)$: $res[i] = 1$

else: $res[i] = f(i-1) + f(i-2)$

return $res[i]$

Зачем это

рес, так

каждому числу

можно без рекурсии

$$\text{res}[1] = \text{res}[2] = 1$$

for ($i = 3 \dots n$):

$O(n)$

$$\text{res}[i] = \text{res}[i-1] + \text{res}[i-2]$$



$$\left. \begin{array}{l} \boxed{\text{~~~~~} i-1} i \leftarrow \text{dp}[i-1] \\ \boxed{\text{~~~~~} i-2} i \leftarrow \text{dp}[i-2] \end{array} \right\} \text{dp}[i]$$

$\text{dp}[i]$: # crickets going to i

$$\text{dp}[i] = \text{dp}[i-1] + \text{dp}[i-2]$$



$$\left. \begin{array}{l} \boxed{\text{~~~~~} i-1} i \leftarrow \text{dp}[i-1] \\ \boxed{\text{~~~~~} i-2} i \leftarrow \text{dp}[i-2] \\ \boxed{\text{~~~~~} i-3} i \leftarrow \text{dp}[i-3] \end{array} \right\} \text{dp}[i]$$

$$\text{dp}[i] = \text{dp}[i-1] + \text{dp}[i-2] + \text{dp}[i-3]$$

$$\text{dp}[1] = \text{dp}[2] = 1$$

$$\text{dp}[3] = 2 \quad // \begin{array}{l} 1 \rightarrow 3 \\ 1 \rightarrow 2 \rightarrow 3 \end{array}$$

for ($i = 4 \dots n$) :

$$dp[i] = dp[i-1] + dp[i-2] + dp[i-3]$$

$l \rightarrow k$

$$dp[1] = dp[2]$$

for ($i = 3 \dots n$)

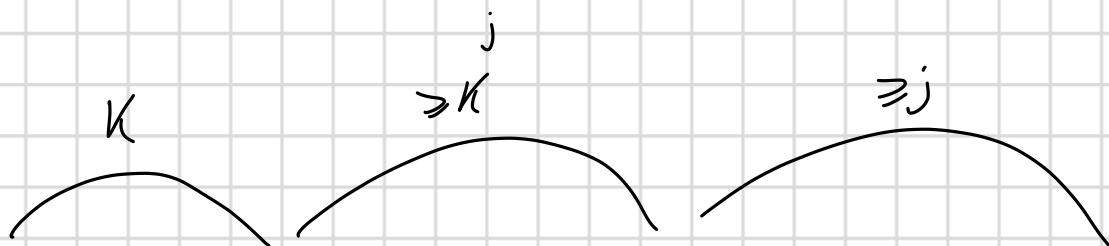
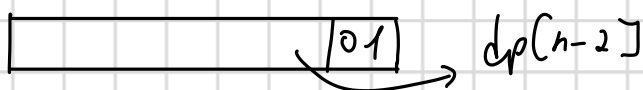
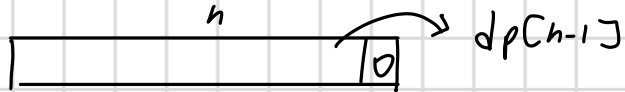
$O(nk)$

for ($j = 1 \dots k$)

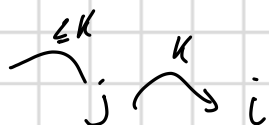
if ($i-j > 0$):

$$dp[i] = dp[i-j]$$

Suma bezprost. dz. "1"



$dp[len][last]$ # encodów podciągu dla $len \leq last$



for ($i = 0 \dots n$)

for ($k = 1 \dots i$)

$O(n^3)$

for ($j = 1 \dots k$)

$$dp[i][k] += dp[i-k][j]$$

Пр:

	1	2	3	4	5	6	7
1	1						
2	1						
3	1	1					
4	1	1	1				
5	1	2		1			
6					1		
7						1	

8 5
↓ ↓
является 2
(5, 2)

3 → 5

↓ ↓

1 → 3 2 → 3

$dp[3][1]$ $dp[3][2]$

(6, 2)

4 → 6 $dp[4][1]$ $dp[4][2]$

(7, 3)

4 → 7

$dp[4][2]$ $dp[4][3]$ $dp[5][1]$

Тогда $[k]$

$[k]$

$dp[i-k][k]$

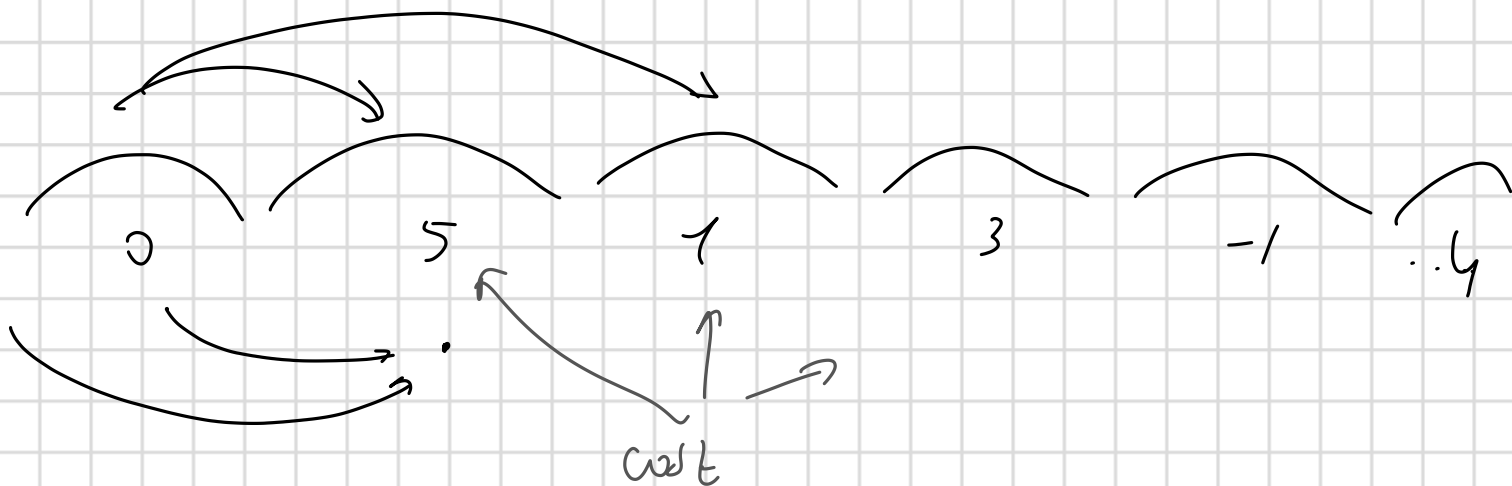
$[k-1]$

$= dp[i][k-1]$

$dp[i][k]$ - # способов получить $\leq k$

$O(n^2)$

$$dp[i][k] = dp[i][k-1] + dp[i-k][k]$$



$$dp[1] = 0$$

$$dp[2] = cost[2]$$

for ($i = 3 \dots n$)

$$dp[i] = \min(dp[i-2], dp[i-1]) + cost[i]$$

$$dp = [0, 5, 1, 4, 0, 4]$$

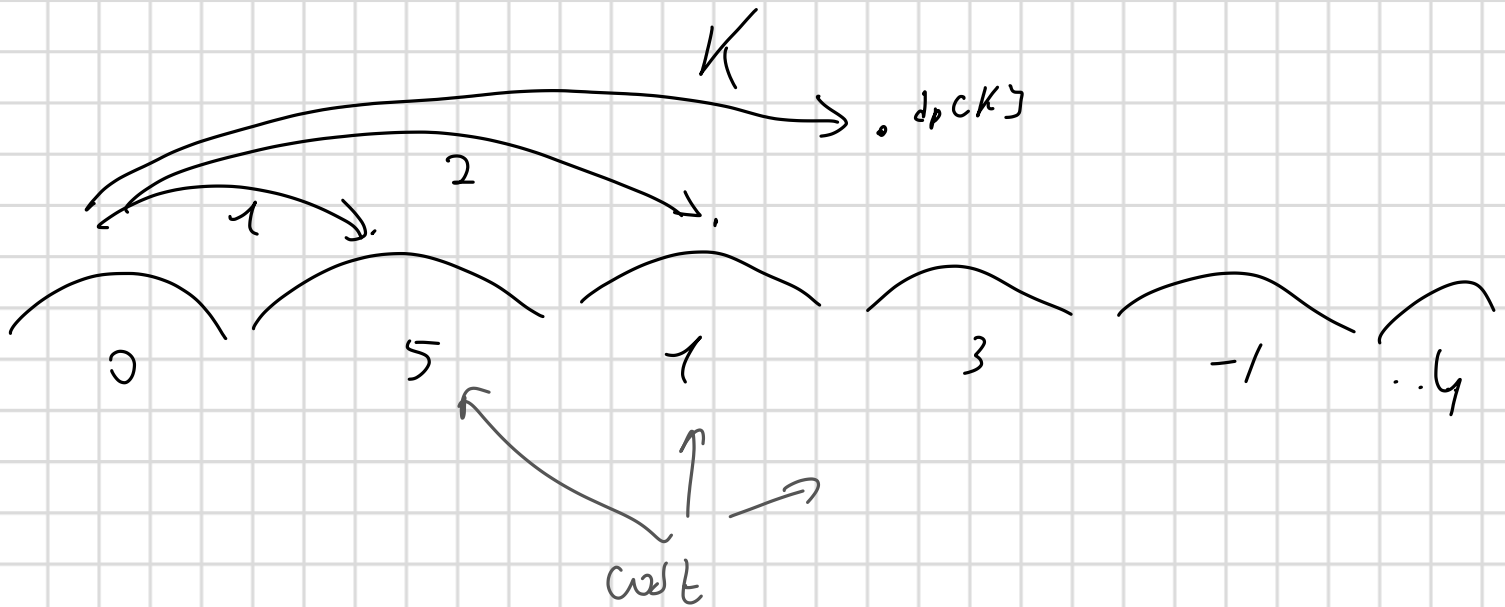
int end = n $dp[0] = \infty$

while (end > 1)

if ($dp[end-1] > dp[end-2]$)

 end -= 2

else: end = 1



$$dp[1] = 0$$

for ($i = 2 \dots n$)

$$dp[i] = \infty$$

for ($j = 1 \dots k$)

if $(i-j < 1)$ break

$$dp[i] = \min(dp[i-1], dp[i-j] + cost[i])$$