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

$f(0) = 0$

$f(1) = 1$

$F_1(n)$:

if $(n<2)$ return n

else return $F_1(n-1) + F_1(n-2)$

Algo1(n)

if $(n<2)$ return n

~~else~~ if $(dp(n) \geq 0)$ return $dp(n)$

else:

$dp(n) = \text{Algo1}(n-1) + \text{Algo1}(n-2)$

return $dp(n)$

// dp[M+1]:

// dp(n) almacena valor F(n).

por defecto es $\boxed{-1}$

· vectorents $dp(n-1)$

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

Algo2(n):

- $dp(0) = 0$

  $dp(1) = 1$

  for $i = 2$ To n:

  $dp(i) = dp(i-1) + dp(i-2)$

  return $dp(n)$

Algo (n)

if (dp(n) < 0) return dp(n)

else

$dp(n) = Algo(n-1) + Algo(n-2)$

return dp(n)

Algo(n):

if (n<2): return n

if (dp(n) != -1) return dp(n)

else

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

return dp(n)

$C(n,k)$ $\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$ $\sim$ $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

dp(n)(k): "Cantidad de formas de escoger k objetos de ~ n"

$C(n,k) = C(n-1,k-1) + C(n-1,k)$

Algo1(n,k)

if (k == 0 || k == n):

return 1

else if (dp(n)(k) != -1)

return dp(n)(k)

else:

$dp(n)(k) = Algo1(n-1, k-1) + Algo1(n-1, k)$

return dp(n)(k)

Algo2(n,k)

for i=0 To n:

dp(n)(0) = dp(n)(n) = 1

for i=1 To n:

for j=1 To i-1:

dp(i)(j) = dp(i-1)(j-1) + dp(i-1)(j)

return dp(n)(k).