I CONIGLI SI RIPRODUCONO DOPO 1 ANNO

$$F = \begin{cases} 1 & \text{SE } m = 1,2 \\ F_{m-1} + F_{m-2} & \text{SE } m \geq 3 \end{cases}$$

COME CALCOLIAMO F_m DATO UN NUMERO m?

FIBONACCI 1

FIBINONACCI1 (m):

RETURN $1/\sqrt{5} \cdot (\phi^m - \bar{\phi}^m)$ → NON È CORRETTO

FIBONACCI 2

FIBONACCI2 (m):

IF (m ≤ 2) THEN RETURN 1

ELSE RETURN FIBONACCI(m-1) + FIBONACCI(m-2) → DIVIDI ET IMPERA

CONTO IL NUMERO DI RIGHE DI CODICE ESEGUITE PER "CALCOLARE" IL COSTO DI TEMPO

PER FIBONACCI 2

$m = 1$ # 1 RIGA

$m = 2$ # 1

$m = 3$ # 4

m GENERICO # $f(m)$?

EQUAZIONE DI RICORRENZA PER FIB2

$T(n)$ = NUMERO DI LINEE DI CODICE ESEGUITE NEL CASO PEGGIORE SU INPUT n.

PER FIB2: $T(n) = 2 + T(n-1) + T(n-2)$ $\qquad T(1) = T(2) = 1$

$\forall n > 2$

IF (prima riga)
ELSE (SECONDA RIGA)

USIAMO L'ALBERO DI RICORRENZA

NODI INTERNI = ETICHETTATI CON 2

FOGLIE = ETICHETTATI CON 1

lemma 1 !!

IL NUMERO DI FOGLIE DI FIB2(n) È PARI A F_n

- OGNI FOGLIA ESEGUE ※ 1 LINEA DI CODICE (IF)

lemma 2

IL NUMERO DI NODI DI FIB2(n) È PARI AL NUM DI FOGLIE -1

- OGNI NODO ESEGUE ※ 2 LINEE DI CODICE (IF - ELSE)

$$T(n) = F_n + \underbrace{2(F_n - 1)} = 3F_n - 2 \longrightarrow F_n \approx \Phi^n$$

LINEE ESEGUITE DALLE FOGLIE SAREBBE $1 \cdot F_n$

NUM. DI LINEE DI CODICE ESEGUITE DALLE FOGLIE

NUMERO DI FOGLIE

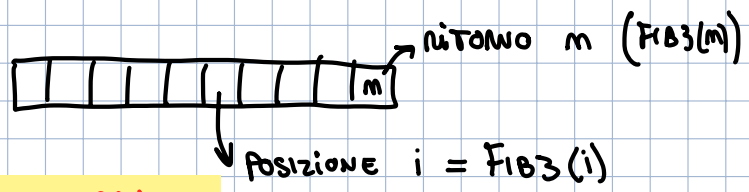
LINEE ESEGUITE DAI NODI $2 \cdot F_n - 1$

NUM. DI LINEE DI CODICE ESEGUITE DAI NODI

NUMERO DI NODI

Fibonacci 3

```
1 algoritmo fibonacci3(intero n) → intero
      sia Fib un array di n interi
2     Fib[1] ← 1; Fib[2] ← 1
3     for i = 3 to n do
4         Fib[i] ← Fib[i-1] + Fib[i-2]
5     return Fib[n]
```



TEMPO = O(m)
MEMORIA = O(m)

- Linee 1, 2, e 5 eseguite una sola volta
- Linea 3 eseguita $\leq n$ volte
- Linea 4 eseguita $\leq n$ volte
- T(n): numero di linee di codice mandate in esecuzione da `fibonacci3`

$T(n) \leq n + n + 3 = 2n + 3$

$T(45) \leq 93$

38 milioni di volte più veloce dell'algortimo `fibonacci2`!

LINEA 1, 2, 5 SONO ESEGUITE UNA SOLA VOLTA

LINEA 3, 4 SONO ESEGUITE $\leq$ m VOLTE

$T(m) \leq m + m + 3 = \boxed{2m + 3}$ ✓ $\boxed{O(m)}$!!

Fibonacci 4

USA MENO MEMORIA DI FIBONACCI 3

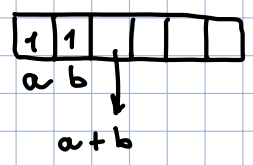
```
algoritmo fibonacci4(intero n) → intero
    a ←1, b ←1
    for i = 3 to n do
        c ← a+b
        a ← b
        b ← c
    return c
```

$T(n) \leq 4n + 2$

più veloce o più lento di `fibonacci3`?

NAH...

POSSO CALCOLARE F_i CON F_{i-1} + F_{i-}



TEMPO = O(m)
MEMORIA = O(1) COSTANTE

$\boxed{O(m)}$!!

QUESTO PERCHÉ STO CONTANDO LE LINEE DI CODICE, QUINDI É UN MODO GROSSOLANO

Usiamo la notazione asintotica

- IGNORO COSTANTI MOLTIPLICATIVE
- IGNORO TERMINI DI ORDINE INFERIORE

$T(m) = 5m + 9 = O(m)$

$T(m) = 6m^2 - 9 + 18 = O(m^2)$

FIBONACCI 5

POSSO RICAVARE F_m USANDO QUESTA PROPRIETÀ:

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$$

CHE NELLA DIM. PER INDUZIONE, POSSIAMO RAPPRESENTARE IL CASO INDUTTIVO COME:

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \overbrace{\begin{pmatrix} F_n & F_{n-1} \\ F_{n-1} & F_{n-2} \end{pmatrix}}^{m-1} \overbrace{\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}}^{m}$$

$$= \begin{pmatrix} F_n + F_{n-1} & F_n \\ F_{n-1} + F_{n-2} & F_{n-1} \end{pmatrix} = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$$

NOTIAMO CHE NEL PASSO M-1 NELLA POSIZIONE IN ALTO A SX ABBIAMO F_m OVVERO IL NUMERO CHE MI SERVE

algoritmo `fibonacci5`$(intero\ n) \rightarrow intero$

1. $M \leftarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
2. **for** $i = 1$ **to** $n - 1$ **do**
3. $\quad M \leftarrow M \cdot \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$
4. **return** $M[0][0]$

ARRIVIAMO AL PASSO M-1, POTREMMO ARRIVARE ANCHE AL PASSO M MA FAREMMO UN PASSO IN PIÙ E AVREMMO IL RISULTATO IN POSIZIONE DIVERSA $\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$

TEMPO $O(m)$

FIBONACCI 6

POSSIAMO CALCOLARE LA M-ESIMA POTENZA ELEVANDO AL QUADRATO LA $(\lfloor m/2 \rfloor)$ - ESIMA POTENZA:

$$\overset{m}{} \quad \lfloor m/2 \rfloor \quad \text{ELEVAZIONE AL QUADRATO}$$

$2^4 = (2^2)^2$ $\qquad 2^8 = (2^4)^2 = ((2^2)^2)$ $\qquad 2^5 = (2^2)^2 \cdot 2$

ESEGUENDO UNA ULTERIORE MOLTIPLICAZIONE SE m È DISPARI.

SAPPIAMO CHE POSSIAMO SFRUTTARE QUESTA FORMULA CHIUSA PER CALCOLARE FIBONACCI:

$F_m = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^m$ CHE PER LA DEF. DI SOPRA SAREBBE $F_m = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^m = \left(\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{m/2}\right)^2$

algoritmo `fibonacci6`($intero\ n$) $\rightarrow$ $intero$
  $A \leftarrow \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$
  $M \leftarrow$ `potenzaDiMatrice`$(A, n-1)$
  return $M[0][0]$

funzione `potenzaDiMatrice`($matrice\ A$, $intero\ k$) $\rightarrow$ $matrice$
  if ($k = 0$) **return** $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
  else $M \leftarrow$ `potenzaDiMatrice`$(A, \lfloor k/2 \rfloor)$
    $M \leftarrow M \cdot M$
  if (k è dispari) **then** $M \leftarrow M \cdot A$
  return M

FIB6

$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$

$M =$
$k = 5$

POTENZA DI MATRICE

$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$	$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$	$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$	$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$
$k = 4$	$k = 2$	$k = 1$	$k = 0$
$M =$	$M =$	$M =$	$M =$
$\lfloor k/2 \rfloor$	$\lfloor k/2 \rfloor$	$\lfloor k/2 \rfloor$	

FIB6

$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$

$M =$
$k = 5$

RETURN $M[0][0]$

$(x)^5 = (x^2)^2 \cdot (x)$

Potenza di matrice

$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$	$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$	$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$	$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$
$K = 4$	$K = 2$	$K = 1$	$K = 0$
$M =$	$M =$	$M =$	$M =$
$M = M \cdot M$	$M = M \cdot M$	$M = M \cdot M$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
		$M = M \cdot A$	
AUDIO (4)	AUDIO (3)	AUDIO (2)	AUDIO (1)

PRENDENDO COME ESEMPIO $x^5 = (x^2)^2 \cdot x$, NELLO STACK DISEGNATO SOPRA SAREBBE:

Potenza di matrice

$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$	$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$	$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$	$A = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$
$K = 4$	$K = 2$	$K = 1$	$K = 0$
$M =$	$M =$	$M =$	$M =$
$M = M \cdot M$	$M = M \cdot M$	$M = M \cdot M$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
		$M = M \cdot A$	
$(x^2)^2$	x^2	(x)	(1)

RISCRIVIAMO LE POTENZE SECONDO FIBONACCI6

```
Global frame
funzionePotenza        function funzionePotenza(numero, potenza)

funzionePotenza
        numero   2
        potenza  7

funzionePotenza
        numero   2
        potenza  3

funzionePotenza
        numero   2
        potenza  1

funzionePotenza
        numero   2
        potenza  0
        Return value  1
```

$$2^7 = 2^6 \cdot 2 = 2^3 \cdot 2^3 \cdot 2 = \underbrace{2^2 \cdot 2}_{2^3} \cdot \underbrace{2^2 \cdot 2}_{2^3} \cdot 2$$

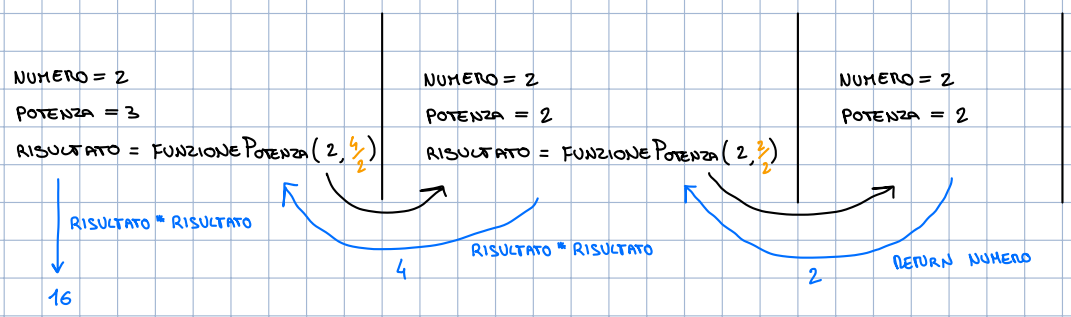
$$2^6 = 2^3 \cdot 2^3$$

$$2^3 = 2^2 \cdot 2$$

Esempio

```python
def funzionePotenza(numero,potenza):

    if(potenza <= 1):
        return numero

    else:
        risultato = funzionePotenza(numero,potenza//2)
        risultato = risultato * risultato

    if(potenza%2 == 1):
        risultato = risultato * numero

    return risultato
```

FUNZIONE POTENZA (2, 4);



QUESTA FUNZIONE POSSIEDE UNA VARIABILE CHIAMATA NUMERO CHE RAPPRESENTA IL NUM. DA ELEVARE A POTENZA, LA VARIABILE POTENZA CHE RAPPRESENTA IL NUMERO DI VOLTE PER CUI MOLTIPLICARE IL NUMERO PER SE STESSO.

CON QUESTA FUNZIONE UTILIZZIAMO LA PROPRIETÀ POTENZA DI POTENZE COME AD ESEMPIO: $2^4 = (2^2)^2$ OVVERO $x^m = \left(x^{\frac{m}{2}}\right)^2$

LA NOSTRA FUNZIONE IN OGNI CHIAMATA INVOCA RICORSIVAMENTE SE STESSA PASSANDO COME PARAMETRI IL NUMERO E LA POTENZA DIMEZZATA, ALL' INTERNO DI OGNI CHIAMATA:

- SE POTENZA == 1 ⟶ RITORNA NUMERO
- ALTRIMENTI ⟶ RISULTATO = RISULTATO * RISULTATO

QUINDI È COME SE FACESSE IL QUADRATO DEL NUMERO CORRENTE, DOPODICHÉ TORNERÀ ALLA CHIAMATA SUPERIORE IL QUADRATO EFFETTUATO, LA CHIAMATA SUPERIORE PRENDERÀ QUEL VALORE E LO ELEVERÀ AL QUADRATO, FACENDO QUINDI IL QUADRATO DEL QUADRATO.

PERCHÉ L'ULTIMA CHIAMATA RITORNA IL NUMERO?

VISTO CHE OGNI CHIAMATA EFFETTUA IL QUADRATO DEL NUMERO RITORNATO DALLA CHIAMATA SOTTOSTANTE, L'ULTIMA CHIAMATA RITORNA IL NUMERO STESSO COSÍ CHE LA PENULTIMA CHIAMATA FARÀ IL QUADRATO DEL NUMERO RICEVUTO DALL' ULTIMA CHIAMATA, CHE SARÀ PROPRIO IL NUMERO INIZIALE, FACENDO COSÍ IL QUADRATO DEL NUMERO INIZIALE.

ATTENZIONE SI NOTI L'ESEMPIO CON UNA POTENZA DISPARI.