

Ricorsione

Algoritmi e Strutture Dati

Luciano Margara

Unibo

2022

Fattoriale

FATTORIALE(n)

1 if $n == 1$

2 return 1

3 else

4 return FATTORIALE($n - 1$) * n

Numeri di Fibonacci

FIB(n)

1 if $n == 1$

2 return 1

3 if $n == 2$

4 return 1

5 return FIB($n - 1$) + FIB($n - 2$)

Elevamento a potenza

POTENZA(a, n)

1 if $n == 1$

2 return a

3 else

4 return POTENZA($a, n - 1$) * a

Inversione di una lista

INVERTI-LISTA(l)

```
1  if  $l == \text{NIL}$  or  $l.\text{next} == \text{NIL}$ 
2      return  $l$ 
3  else
4       $y = \text{INVERTI-LISTA}(l.\text{next})$ 
5       $l.\text{next}.\text{next} = l$ 
6       $l.\text{next} = \text{NIL}$ 
7      return  $y$ 
```

Ordinamento

`SORT(V)`

```
1  if V.length == 1
2      return V
3  else
4      L = SORT(prima metà di V)
5      R = SORT(seconda metà di V)
6      S = MERGE(L, R)
7      return S
```

Ricorsione: elementi fondamentali

- ▷ Casi base
- ▷ Chiamate ricorsive
- ▷ Passo ricorsivo

Ordinamento: casi base

`SORT(V)`

1 **if** $V.length == 1$

2 **return** V

3 **else**

4 $L = \text{SORT}(\text{prima metà di } V)$

5 $R = \text{SORT}(\text{seconda metà di } V)$

6 $S = \text{MERGE}(L, R)$

7 **return** S

Ordinamento: chiamate ricorsive

`SORT(V)`

1 if *V.length* == 1

2 return *V*

3 else

4 *L* = SORT(prima metà di *V*)

5 *R* = SORT(seconda metà di *V*)

6 *S* = MERGE(*L*, *R*)

7 return *S*

Ordinamento: passo ricorsivo

`SORT(V)`

1 if *V.length* == 1

2 return *V*

3 else

4 *L* = SORT(prima metà di *V*)

5 *R* = SORT(seconda metà di *V*)

6 *S* = MERGE(*L*, *R*)

7 return *S*

Ricorsione corretta: regole

- ▷ I casi base devono essere risolti correttamente
- ▷ Qualunque sia l'input, le catene delle chiamate ricorsive devono essere ben fondate ovvero devono sempre arrivare a un caso base
- ▷ Assumendo corretti i risultati delle chiamate ricorsive, il passo ricorsivo deve produrre una soluzione corretta

Regole applicate al calcolo del fattoriale

- ▷ Casi base: il fattoriale di 1 è effettivamente uguale a 1
- ▷ Catene della ricorsione ben fondate: per qualsiasi valore di n le chiamate ricorsive arrivano sempre a un caso base
- ▷ Passo ricorsivo: se il fattoriale di $n - 1$ è calcolato correttamente allora anche il fattoriale di n è calcolato correttamente come $fattoriale(n - 1) * n$

Numeri di Fibonacci: casi base

$\text{FIB}(n)$

1 **if** $n == 1$

2 **return** 1

3 **if** $n == 2$

4 **return** 1

5 **return** $\text{FIB}(n - 1) + \text{FIB}(n - 2)$

Numeri di Fibonacci: chiamate ricorsive

$\text{FIB}(n)$

1 **if** $n == 1$

2 **return** 1

3 **if** $n == 2$

4 **return** 1

5 **return** $\text{FIB}(n - 1) + \text{FIB}(n - 2)$

Numeri di Fibonacci: passo ricorsivo

$\text{FIB}(n)$

1 if $n == 1$

2 return 1

3 if $n == 2$

4 return 1

5 return $\text{FIB}(n - 1) + \text{FIB}(n - 2)$

Numeri di Fibonacci: errore nei casi base

$\text{FIB}(n)$

1 **if** $n == 1$

2 **return** 1

3 **if** $n == 2$

4 **return** 2

5 **return** $\text{FIB}(n - 1) + \text{FIB}(n - 2)$

Numeri di Fibonacci: catena delle chiamate ricorsive non ben fondate

$\text{FIB}(n)$

1 **if** $n == 1$

2 **return** 1

3 **return** $\text{FIB}(n - 1) + \text{FIB}(n - 2)$

Numeri di Fibonacci: errore passo ricorsivo

FIB(n)

1 if $n == 1$

2 return 1

3 if $n == 2$

4 return 1

5 return FIB($n - 1$) + FIB($n - 2$)

Ricorsione: come funziona [1/10]

1	...	FATTORIALE(n)
2	$n = 4$	1 sia x una nuova variabile
3	$n = \text{FATTORIALE}(n)$	2 if $n == 1$ return 1
4	PRINT(n)	3 $x = \text{FATTORIALE}(n - 1)$
5	...	4 return($x * n$)

Stack della ricorsione

Memoria di lavoro

$n = \dots$

Ricorsione: come funziona [2/10]

1	...	FATTORIALE(n)
2	$n = 4$	1 sia x una nuova variabile
3	$n = \text{FATTORIALE}(n)$	2 if $n == 1$ return 1
4	PRINT(n)	3 $x = \text{FATTORIALE}(n - 1)$
5	...	4 return($x * n$)

Stack della ricorsione

Memoria di lavoro

$n = 4$

Ricorsione: come funziona [3/10]

1	...	FATTORIALE(n)
2	$n = 4$	1 sia x una nuova variabile
3	$n = \text{FATTORIALE}(n)$	2 if $n == 1$ return 1
4	PRINT(n)	3 $x = \text{FATTORIALE}(n - 1)$
5	...	4 return($x * n$)

Stack della ricorsione

$x = \dots$ $n = 4$

Memoria di lavoro

$n = \dots$

Ricorsione: come funziona [4/10]

1	...	FATTORIALE(n)
2	$n = 4$	1 sia x una nuova variabile
3	$n = \text{FATTORIALE}(n)$	2 if $n == 1$ return 1
4	PRINT(n)	3 $x = \text{FATTORIALE}(n - 1)$
5	...	4 return($x * n$)

Stack della ricorsione	Memoria di lavoro
$x = \dots \quad n = 3$	
$x = \dots \quad n = 4$	$n = \dots$

Ricorsione: come funziona [5/10]

1	...	FATTORIALE(n)
2	$n = 4$	1 sia x una nuova variabile
3	$n = \text{FATTORIALE}(n)$	2 if $n == 1$ return 1
4	PRINT(n)	3 $x = \text{FATTORIALE}(n - 1)$
5	...	4 return($x * n$)

Stack della ricorsione	Memoria di lavoro
$x = \dots \quad n = 2$	
$x = \dots \quad n = 3$	
$x = \dots \quad n = 4$	$n = \dots$

Ricorsione: come funziona [6/10]

1	...	FATTORIALE(n)
2	$n = 4$	1 sia x una nuova variabile
3	$n = \text{FATTORIALE}(n)$	2 if $n == 1$ return 1
4	PRINT(n)	3 $x = \text{FATTORIALE}(n - 1)$
5	...	4 return ($x * n$)

Stack della ricorsione	Memoria di lavoro
$x = \dots \quad n = 1$	
$x = \dots \quad n = 2$	
$x = \dots \quad n = 3$	
$x = \dots \quad n = 4$	$n = \dots$

Ricorsione: come funziona [7/10]

1	...	FATTORIALE(n)
2	$n = 4$	1 sia x una nuova variabile
3	$n = \text{FATTORIALE}(n)$	2 if $n == 1$ return 1
4	PRINT(n)	3 $x = \text{FATTORIALE}(n - 1)$
5	...	4 return($x * n$)

Stack della ricorsione	Memoria di lavoro
$x = 1 \quad n = 2$	
$x = \dots \quad n = 3$	
$x = \dots \quad n = 4$	$n = \dots$

Ricorsione: come funziona [8/10]

1	...	FATTORIALE(n)
2	$n = 4$	1 sia x una nuova variabile
3	$n = \text{FATTORIALE}(n)$	2 if $n == 1$ return 1
4	PRINT(n)	3 $x = \text{FATTORIALE}(n - 1)$
5	...	4 return ($x * n$)

Stack della ricorsione	Memoria di lavoro
$x = 2 \quad n = 3$	
$x = \dots \quad n = 4$	$n = \dots$

Ricorsione: come funziona [9/10]

1	...	FATTORIALE(n)
2	$n = 4$	1 sia x una nuova variabile
3	$n = \text{FATTORIALE}(n)$	2 if $n == 1$ return 1
4	PRINT(n)	3 $x = \text{FATTORIALE}(n - 1)$
5	...	4 return ($x * n$)

Stack della ricorsione

$x = 6$ $n = 4$

Memoria di lavoro

$n = \dots$

Ricorsione: come funziona [10/10]

1	...	FATTORIALE(n)
2	$n = 4$	1 sia x una nuova variabile
3	$n = \text{FATTORIALE}(n)$	2 if $n == 1$ return 1
4	PRINT(n)	3 $x = \text{FATTORIALE}(n - 1)$
5	...	4 return($x * n$)

Stack della ricorsione

Memoria di lavoro

$n = 24$