

La ricorsione

1
lez04(ricorsione)

Diverse politiche

- Top-Down
 - Si parte dal problema generale
 - Poi si scende nei particolari (moduli, funzioni, ecc...)
- Bottom-Up
 - Si creano prima i moduli e le funzioni base
 - Poi si assemblano nel programma generale
- Di solito è meglio alternare le due tecniche

2
lez04(ricorsione)

PROGRAMMAZIONE TOP-DOWN

- In molti casi, il problema dato è troppo difficile per scrivere il programma “di getto”
- Scomponiamo il problema in raffinamenti successivi:
 1. livello di specifica
 2. livello di specifica
 3. ...
- Le funzioni ci aiutano in questa scomposizione
 1. prima scriviamo la funzione principale usando funzioni che definiremo poi
 2. poi definiamo le funzioni usate dalla principale usando altre funzioni,
 3. poi definiamo le funzioni ancora non definite,
 4. ...

3
lez04(ricorsione)

ESEMPIO

```
function principale
x=input('dammi x: ');
q=quadruplo(x);
disp(q);
```

```
function out=quadruplo(x)
out=2*doppio(x);
```

```
function out=doppio(x)
out=2*x;
```

4
lez04(ricorsione)

LA RICORSIONE

- In particolare, anche nella *definizione* di una funzione $f()$ può essere utile riutilizzare la funzione $f()$ stessa
- È lo stesso principio usato in matematica:

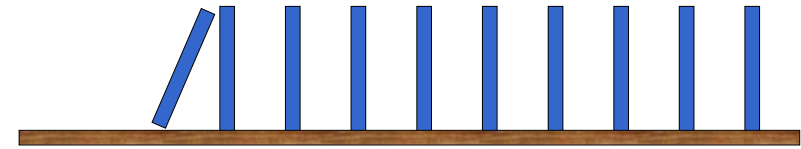
Principio di Induzione

- se una proprietà P vale per $n=n_0$ → CASO BASE
- e si può provare che, *assumendola valida per n*, allora vale per $n+1$
allora P vale per ogni $n \geq n_0$

5

lez04(ricorsione)

IL PRINCIPIO DI INDUZIONE



- Es. ho una sequenza infinita di pezzi del domino disposti come in figura
 - il primo cade
 - Il primo fa cadere il secondo
 - il secondo fa cadere il terzo
 - ...
- Come faccio a dimostrare che cadono tutti?
- Col principio di induzione posso farlo, sapendo che
 - il primo cade
 - se cade l' n -esimo, allora cade l' $(n+1)$ -esimo

6

lez04(ricorsione)

IL PRINCIPIO DI INDUZIONE

Teorema: la somma dei primi n numeri dispari è uguale ad n^2 .

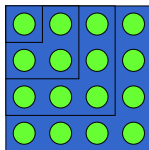
Dimostrazione:

- è vera per $n=1$, infatti $1^2=1$
- supponiamo che per n sia vera, cioè

$$1+3+5+\dots+(2n-1) = n^2$$

e dimostriamo che è vera per $n+1$: sommiamo $(2n+1)$ ad entrambi i termini:

$$1 + 3 + 5 + \dots + (2n-1) + (2n+1) = n^2 + 2n + 1$$



7

lez04(ricorsione)

LA RICORSIONE

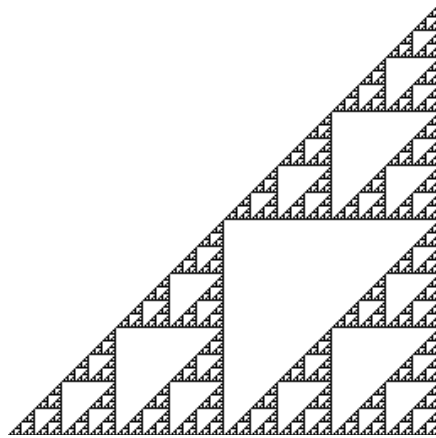
- Possiamo usare il principio di induzione anche per *definire* degli oggetti, non solo per dimostrare proprietà
- In *matematica*, la funzione fattoriale viene definita così:
 - $n! = 1$ se $n=0$
 - $n! = n \cdot (n-1)!$ altrimenti
- Scomponiamo il problema:
 - troviamo un caso semplice, in cui la soluzione è immediata
 - ci riconduciamo a casi precedenti quando la soluzione non è immediata
- Nei *linguaggi di programmazione* moderni, viene fornita la stessa possibilità
- Questo ci permette di scrivere semplicemente algoritmi che sarebbero molto difficili

8

lez04(ricorsione)

LA RICORSIONE

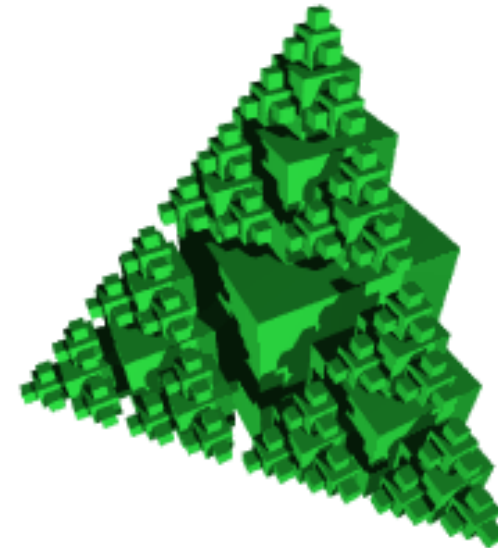
- Pensate a come si potrebbe disegnare la figura a lato (triangolo di Sierpinski)



9

lez04(ricorsione)

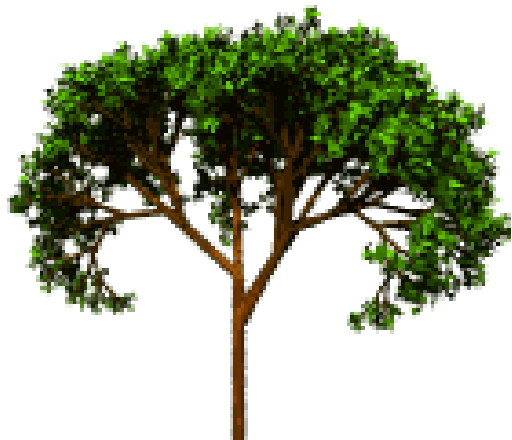
LA RICORSIONE



10

lez04(ricorsione)

LA RICORSIONE



11

lez04(ricorsione)

LA RICORSIONE

- Una funzione matematica è definita *ricorsivamente* quando nella sua definizione compare un riferimento a se stessa
- La ricorsione consiste nella possibilità di *definire una funzione in termini di se stessa*.
- È basata sul *principio di induzione* matematica:
 - se una proprietà P vale per $n=n_0$ \rightarrow CASO BASE
 - e si può provare che, *assumendola valida per n*, allora vale per $n+1$ allora P vale per ogni $n \geq n_0$

12

lez04(ricorsione)

LA RICORSIONE

- Operativamente, risolvere un problema con un approccio ricorsivo comporta
 1. identificare un “caso base” la cui soluzione sia nota
 2. riuscire a esprimere la soluzione al caso generico n in termini dello stesso problema in uno o più casi più semplici ($n-1$, $n-2$, etc).

13
lez04(ricorsione)

LA RICORSIONE: ESEMPIO

Esempio: il fattoriale di un numero

$\text{fact}(n) = n!$

$n! : \mathbb{Z} \rightarrow \mathbb{N}$

$n!$ vale 1 se $n \leq 0$

$n!$ vale $n*(n-1)!$ se $n > 0$

Codifica:

```
function f=fatt(n)
if n<=0
    f=1;
else
    f=n*fatt(n-1);
end
```

14
lez04(ricorsione)

LA RICORSIONE: ESEMPIO

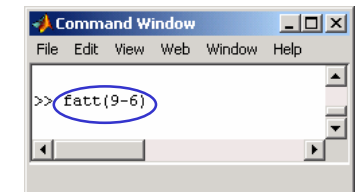
```
function f=fatt(n)
if n<=0
    f=1;
else
    f=n*fatt(n-1);
end
```

15
lez04(ricorsione)

LA RICORSIONE: ESEMPIO

Servitore e cliente

```
function f=fatt(n)
if n<=0
    f=1;
else
    f=n*fatt(n-1);
end
```



Si valuta l'espressione che costituisce il parametro attuale (in command window) e si trasmette alla funzione fatt una copia del valore così ottenuto (3).

16
lez04(ricorsione)

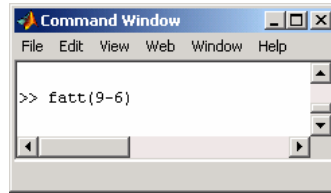
LA RICORSIONE: ESEMPIO

Servitore e cliente

```
function f=fatt(n)
if n<=0
    f=1;
else
    f=n*fatt(n-1);
end
```

3

La funzione `fatt` lega il parametro `n` a 3. Essendo 3 positivo si passa al ramo `else`. Per calcolare il risultato della funzione e' necessario effettuare una nuova chiamata di funzione. `n-1` nell'environment di `fatt` vale 2 quindi viene chiamata `fatt(2)`



17

lez04(ricorsione)

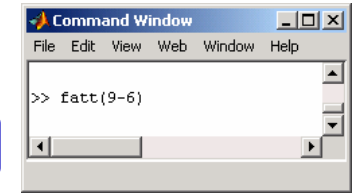
LA RICORSIONE: ESEMPIO

Servitore e cliente

```
function f=fatt(n)
if n<=0
    f=1;
else
    f=n*fatt(n-1);
end
```

2

Il nuovo servitore lega il parametro `n` a 2. Essendo 2 positivo si passa al ramo `else`. Per calcolare il risultato della funzione e' necessario effettuare una nuova chiamata di funzione. `n-1` nell'environment di `fatt` vale 1 quindi viene chiamata `fatt(1)`



18

lez04(ricorsione)

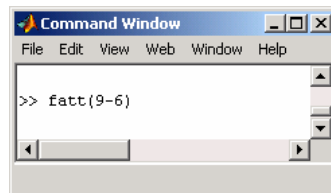
LA RICORSIONE: ESEMPIO

Servitore e cliente

```
function f=fatt(n)
if n<=0
    f=1;
else
    f=n*fatt(n-1);
end
```

1

La funzione `fatt` lega il parametro `n` a 1. Essendo 1 positivo si passa al ramo `else`. Per calcolare il risultato della funzione e' necessario effettuare una nuova chiamata di funzione. `n-1` nell'environment di `fatt` vale 0 quindi viene chiamata `fatt(0)`



19

lez04(ricorsione)

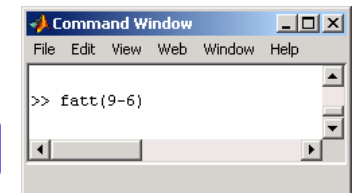
LA RICORSIONE: ESEMPIO

Servitore e cliente

```
function f=fatt(n)
if n<=0
    f=1;
else
    f=n*fatt(n-1);
end
```

0

Il nuovo servitore lega il parametro `n` a 0. La condizione `n <= 0` e' vera e la funzione `fatt(0)` restituisce come risultato 1.



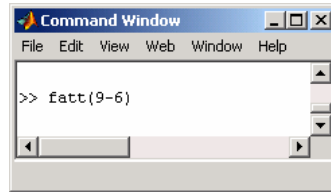
20

lez04(ricorsione)

LA RICORSIONE: ESEMPIO

Servitore e cliente

```
function f=fatt(n)
if n<=0
    f=1;
else
    f=n*fatt(n-1);
end
```



1

1

*Il controllo torna al servitore precedente fatt(1) che puo' valutare l'espressione n * 1 (valutando n nel suo environment dove vale 1) ottenendo come risultato 1.*

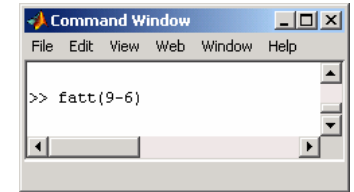
21

lez04(ricorsione)

LA RICORSIONE: ESEMPIO

Servitore e cliente

```
function f=fatt(n)
if n<=0
    f=1;
else
    f=n*fatt(n-1);
end
```



2

1

*Il controllo torna al servitore precedente fatt(2) che puo' valutare l'espressione n * 1 (valutando n nel suo environment dove vale 2) ottenendo come risultato 2.*

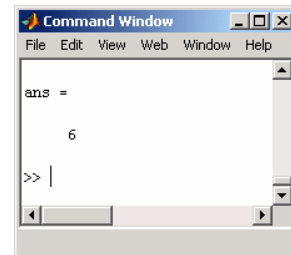
22

lez04(ricorsione)

LA RICORSIONE: ESEMPIO

Servitore e cliente

```
function f=fatt(n)
if n<=0
    f=1;
else
    f=n*fatt(n-1);
end
```



3

2

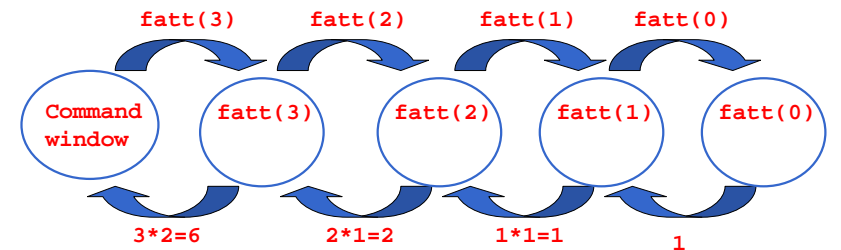
*Il controllo torna al servitore precedente fatt(3) che puo' valutare l'espressione n * 2 (valutando n nel suo environment dove vale 2) ottenendo come risultato 6.*

IL CONTROLLO TORNA ALLA COMMAND WINDOW CHE VISUALIZZA IL VALORE 6

23

lez04(ricorsione)

LA RICORSIONE: ESEMPIO



Command window	fatt(3)	fatt(2)	fatt(1)	fatt(0)
Cliente di fatt(3)	Cliente di fatt(2)	Cliente di fatt(1)	Cliente di fatt(0)	Servitore di fatt(1)
	Servitore della CW	Servitore di fatt(3)	Servitore di fatt(2)	

24

lez04(ricorsione)

LA RICORSIONE: ESEMPIO

Problema:
calcolare la somma dei primi N interi

Specifica:

Considera la somma $1+2+3+\dots+(N-1)+N$ come composta di due termini:

- $(1+2+3+\dots+(N-1))$ → *Il primo termine non è altro che lo stesso problema in un caso più semplice: calcolare la somma dei primi N-1 interi*
- N → *Valore noto*

Esiste un caso banale ovvio: CASO BASE

- la somma fino a 1 vale 1.

25

lez04(ricorsione)

LA RICORSIONE: ESEMPIO

Problema:
calcolare la somma dei primi N interi

Algoritmo ricorsivo

Se N vale 1 allora la somma vale 1

altrimenti la somma vale N + il risultato della somma dei primi N-1 interi

26

lez04(ricorsione)

LA RICORSIONE: ESEMPIO

Problema:
calcolare la somma dei primi N interi

Codifica:

```
function s=sommafinoa(n)
if n==1;
    s=1;
else
    s=n+sommafinoa(n-1);
end
```

27

lez04(ricorsione)

Ricerca di un elemento in un vettore ordinato

- Sapendo che il vettore è *ordinato*, la ricerca può essere ottimizzata.
 - *Vettore ordinato in senso non decrescente:*
 - Esiste una relazione d'ordine totale sul dominio degli elementi del vettore e:
 - Se $i < j$ si ha $v(i) \leq v(j)$
- | | | | | | | | |
|---|---|---|---|---|---|----|----|
| 2 | 3 | 5 | 5 | 7 | 8 | 10 | 11 |
|---|---|---|---|---|---|----|----|
- *Vettore ordinato in senso crescente:*
 - Se $i < j$ si ha $v(i) < v(j)$
- | | | | | | | | |
|---|---|---|---|---|---|----|----|
| 2 | 3 | 5 | 6 | 7 | 8 | 10 | 11 |
|---|---|---|---|---|---|----|----|
- In modo analogo si definiscono l'ordinamento in senso *non crescente* e *decrescente*.

28

lez04(ricorsione)

RICERCA BINARIA

- Ricerca binaria di un elemento in un vettore ordinato in senso non decrescente in cui il primo elemento e' **first** e l'ultimo **last**.
- La tecnica di *ricerca binaria* rispetto alla ricerca esaustiva, consente di eliminare ad ogni passo metà degli elementi del vettore.

29
lez04(ricorsione)

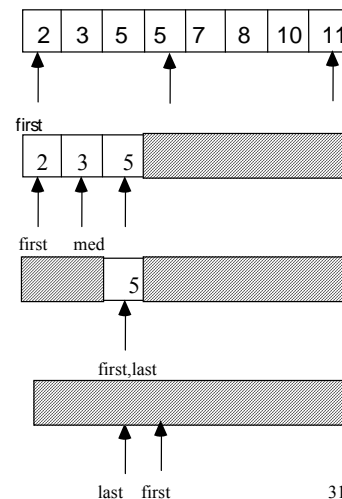
RICERCA BINARIA

- Se **first** ≤ **last**
 - Confronta l'elemento cercato **e1** con quello mediano del vettore, **V(med)**.
 - Se **e1 == V(med)**, fine della ricerca: trovato!
 - Altrimenti,
 - se **e1 < V(med)**, ripeti la ricerca nella prima metà del vettore (indici da **first** a **med-1**);
 - se **e1 > V(med)**, ripeti la ricerca nella seconda metà del vettore (indici da **med+1** a **last**).

30
lez04(ricorsione)

RICERCA BINARIA

- Esempio: si cerca il valore **e1=4**
- **med=floor((first+last)/2)**
- **e1 < V(med)**
- **e1 > V(med)**
- **e1 < V(med)**
- **fine**



31
lez04(ricorsione)

RICERCA BINARIA (Codice)

```
function ricerca(A,x)
%ricerca binaria
first=1;
last=length(A);
disp(A);
if first<=last
    med=floor((first+last)/2);
    if A(med)==x
        disp('trovato!!!')
    elseif A(med)>x
        ricerca(A(1:med-1),x);
    elseif A(med)<x
        ricerca(A(med+1:end),x);
    end
else
    disp('non trovato');
end
```

32
lez04(ricorsione)