Appunti di informatica

Lezione 13

anno accademico 2016-2017

Mario Verdicchio

Ricorsione

Un approccio alla programmazione

Ricorsione: suoi componenti

- Caso base
- Ipotesi ricorsiva
- Passo
- Integrazione

Idea di base della ricorsione

- Siamo di fronte a un problema P
- Ci chiediamo se il problema P abbia una soluzione immediata (caso base)
  - In tal caso, il problema è risolto
- Altrimenti, cerchiamo di scomporre il problema P in due parti:
  - Il sottoproblema P' che ha la stessa struttura del problema iniziale P, ma si applica a un input diverso: in particolare un input più vicino al caso base
  - Il passo, ovvero quell'operazione che permette di creare la soluzione del problema P a partire dalla soluzione al problema P'
- L'ipotesi ricorsiva è la supposizione sul fatto che abbiamo già una soluzione per il problema P'
- L'integrazione è il processo di composizione della soluzione finale

Esempio: il calcolo del fattoriale di n

- Il fattoriale di un numero intero nullo o positivo n è definito come segue:

  n! = n*(n-1)*(n-2)*(n-3)*…*3*2*1

  es. 6! = 6*5*4*3*2*1 = 720

  per definizione, 0! = 1
- Problema P: dato in input un numero n, calcolarne il fattoriale n!
- Soluzione ricorsiva:
  - Controllo se sono nel caso base, ovvero, controllo se n == 0
  - Se non sono nel caso base, scompongo P in P' (sottoproblema con la stessa struttura ma input diverso) e il passo:
    - n! = n*(n-1)!

n! = n*(n-1)!

- L'ipotesi ricorsiva equivale a supporre di conoscere il valore di (n-1)!
- Il raggiungimento della soluzione di P si ottiene eseguendo il passo: moltiplichiamo il risultato di (n-1)! per n.

Integrazione

```
def fatt(n):
        if n==0:
                return 1
        return n*fatt(n-1)
```

Uso nel programma

```
from __future__ import print_function
def fatt(n):
        if n==0:
                return 1
        return n*fatt(n-1)
while True:
        x = input("dammi un numero positivo o nullo\n")
        if x >= 0:
                break
print("il suo fattoriale e':", fatt(x))
```

Altro problema: palindromia delle stringhe

- Una stringa di caratteri si dice «palindroma» quando si legge allo stesso modo da sinistra e da destra
- Ad esempio: «abba» oppure «itopinonavevanonipoti» sono palindrome, mentre non lo è «casa»
- Scrivere una funzione ricorsiva in Python che, presa una stringa in input, dica con un booleano (True, False) se essa è palindroma o meno.

Integrazione

```
def pal(str):
        if len(str)<=1:
                return True
        return str[0]==str[len(str)-1] and pal(str[1:len(str)-1])
```

Uso nel programma

```
from __future__ import print_function
def pal(str):
        if len(str)<=1:
                return True
        return str[0]==str[len(str)-1] and pal(str[1:len(str)-1])
stringa = raw_input("dammi una stringa\n")
if (pal(stringa)==True):
        print("la stringa e' palindroma")
else:
        print("la stringa non e' palindroma")
```

La serie di Fibonacci

- Si tratta di una sequenza di numeri interi positivi come segue:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,…

- Scrivere una funzione ricorsiva che, dato in input un intero positivo n, restituisca l'n-esimo numero della serie di Fibonacci.

Integrazione

```
def fib(n):
        if n<=2:
                return 1
        return fib(n-1) + fib(n-2)
```