

Laurea Triennale in Ingegneria Gestionale

Corso di Fondamenti di Informatica A.A. 2017/2018

DEPARTMENT OF COMPUTER, CONTROL, AND
MANAGEMENT ENGINEERING ANTONIO RUBERTI



SAPIENZA
UNIVERSITÀ DI ROMA

Ricorsione

Definizione

Ricorsione

- una tecnica per risolvere problemi **scomponendo** un problema difficile in problemi più semplici finché non si arriva ad un problema elementare che si può risolvere molto facilmente (problema banale)
- La ricorsione di solito si manifesta da una funzione che chiama se stessa(!)

Perché?

- Tanti problemi possono essere risolti in maniera semplice ed elegante usando la ricorsione

Esempio: sommalista

Problema: calcolare la somma dei numeri contenuti in una lista

Ad esempio, se la lista è:

$$l = [1, 3, 5, 7, 9]$$

La somma dei numeri contenuti è

$$1+3+5+7+9 = 25$$

Esempio: sommalista (1/5)

Soluzione #1 (senza ricorsione):

```
def sommalista(l):  
    somma = 0  
    for numero in l:  
        somma += numero  
    return somma
```

Esempio: sommalista (2/5)

Si può risolvere il problema senza l'uso di un ciclo `for` o `while` considerando che il problema si può scomporre nel modo seguente:

$$somma = (1 + (3 + (5 + (7 + 9))))$$

$$somma = (1 + (3 + (5 + 16)))$$

$$somma = (1 + (3 + 21))$$

$$somma = (1 + 24)$$

$$somma = 25$$

Esempio: sommalista (3/5)

Osservazione: in ogni coppia di parentesi abbiamo la somma di un numero con la somma di tutti gli altri numeri:

$$somma = (1 + (3 + (5 + (7 + 9))))$$

$$somma = (1 + (3 + (5 + 16)))$$

$$somma = (1 + (3 + 21))$$

$$somma = (1 + 24)$$

$$somma = 25$$

Abbiamo suddiviso il problema in problemi più semplici:

- ad ogni riga la parentesi più annidata contiene solo la somma fra due numeri

Esempio: sommalista (4/5)

In ogni coppia di parantesi abbiamo la somma di un numero con la somma di tutti gli altri numeri:

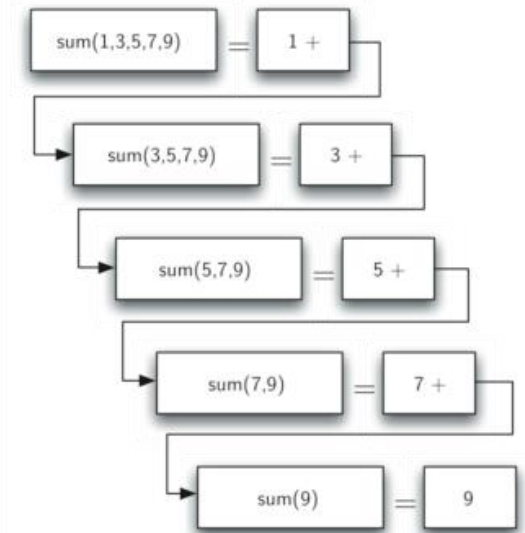
$$somma = (1 + (3 + (5 + (7 + 9))))$$

$$somma = (1 + (3 + (5 + 16)))$$

$$somma = (1 + (3 + 21))$$

$$somma = (1 + 24)$$

$$somma = 25$$



Quindi se la lista ha duo o più elementi abbiamo:

```
sommalista(l) = l[0] + sommalista(l[1:len(l)])
```

e se ne ha solo uno:

```
sommalista(l) = l[0]
```

Esempio: sommalista (5/5)

Soluzione #2 (ricorsione):

```
def sommalista(l):  
    if len(l) == 1:  
        return l[0]  
    else:  
        return l[0] + sommalista(l[1:])
```


Le tre regole della ricorsione

1. L'algoritmo ha una condizione di base
2. Ad ogni passo l'algoritmo si deve muovere verso la condizione di base
3. L'algoritmo deve ricorsivamente chiamare se stesso

```
def sommalista(l):  
    if len(l) == 1:  
        return l[0]  
    else:  
        return l[0] + sommalista(l[1:])
```

Condizione di base (Regola 1)

Ricorsione (Regola 3)

Suddivisione del problema (Regola 2)