

11. Ricorsione

Corso di Algoritmi e Linguaggi di Programmazione Python/C

Outline

- Ricorsione vs. Iterazione
- Il calcolo del fattoriale – Approccio iterativo
- Il calcolo del fattoriale – Approccio ricorsivo

Ricorsione vs. Iterazione

- Un metodo **iterativo** ripete un'istruzione (o una serie di istruzioni) fino a che è verificata una condizione.
- Un metodo **ricorsivo** agisce scomponendo un problema in parti sempre più piccole che vengono risolte individualmente.
- L'esecuzione si ferma quando si raggiunge un caso **base**, ovvero una parte non ulteriormente scomponibile.

Il calcolo del fattoriale – Approccio iterativo

- Il fattoriale di un numero è definito come:

$$n! = n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1$$

- Ad esempio, $3! = 6$, $4! = 24$, $5! = 120$, ...
- Un'implementazione **iterativa** del calcolo del fattoriale è proposta in questa slide.

```
fatt = 1;  
for i che varia da 1 ad n:  
    fatt = fatt * i;  
endfor  
return fatt;
```

Il calcolo del fattoriale – Approccio ricorsivo

- Notiamo che:

$$n! = n \cdot (n - 1)! \Rightarrow$$

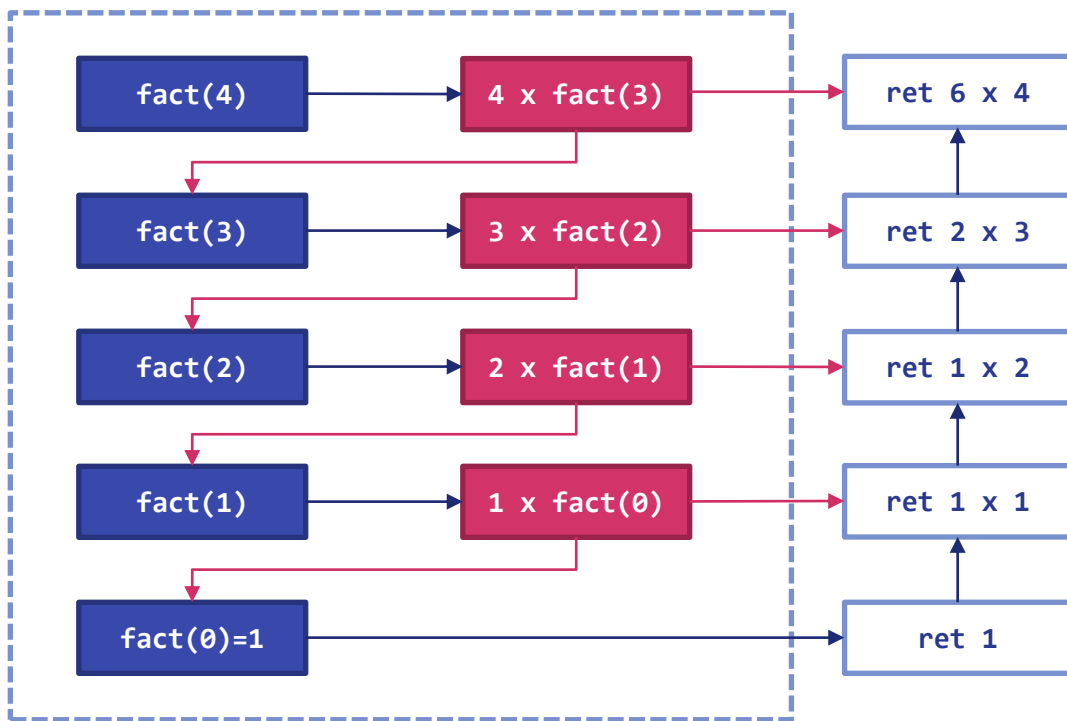
$$\Rightarrow n! = n \cdot (n - 1) \cdot (n - 2)! \Rightarrow$$

$$\Rightarrow \dots \Rightarrow$$

$$\Rightarrow n! = n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1 \cdot 0!$$

- In questa formulazione, scomponiamo il problema iniziale in una serie di sotto-problemi.
 - Se $n > 0$, allora dobbiamo moltiplicare n per $(n - 1)!$
 - Se $n = 0$, non possiamo ulteriormente scomporre il problema.
 - Abbiamo raggiunto il **caso base** per la ricorsione.

Il calcolo del fattoriale – Approccio ricorsivo



Call stack

```
int factorial(int n):  
    if n = 0:  
        return 1;  
    else:  
        return n * factorial(n - 1);  
    endif
```

Domande?

42