

PROGRAMMAZIONE RICORSIVA

[1]

○ L'IDEA BASE

- ↪ Non si basa su cicli (iterativa) ma richiama la stessa funzione piu' volte dal suo interno
- ↪ Soluzione di sottoproblemi analoghi ma piu' semplici

○ ESEMPIO: LUNGHEZZA DI UNA LISTA

ITERATIVA

```
int lung(lista_pt L)
{
    int lunghezza=0;
    while(L!=NULL)
    {
        L=L->Next;
        lunghezza=lunghezza+1;
    }
    return lunghezza;
}
```

RICORSIVA

```
int lung(lista_pt L)
{
    if (L==NULL)
        return 0;
    else
        return (1+lung(L->Next));
}
```

PROGRAMMAZIONE RICORSIVA

[2]

○ CARATTERISTICHE FONDAMENTALI

- ↪ Ogni funzione ricorsiva deve avere almeno un caso non ricorsivo (**argomento terminale**), altrimenti non si ha terminazione
- ↪ Lo scopo del caso ricorsivo e' calcolare il valore finale della funzione a partire dalla chiamata della funzione stessa su un argomento piu' vicino all'argomento terminale
- ↪ Il caso terminale puo' variare da funzione a funzione (tipicamente e' la lista vuota ...)
- ↪ Spesso la ricorsione si basa sulla struttura intrinsecamente ricorsiva delle strutture su cui opera, da cui il nome **ricorsione strutturale**

PROGRAMMAZIONE RICORSIVA

[3]

○ METODOLOGIA PER LA DEFINIZIONE (RICORS. STRUTT.)

- ↗ **Specifica:** descrizione del risultato che si vuole ottenere, una o piu' relazioni in modo formale o testo in linguaggio naturale
- ↗ **Base:** definizione del caso non ricorsivo
- ↗ **Ipotesi ricorsiva:** si suppone che la funzione operi correttamente su un dato piu' semplice
- ↗ **Passo:** come operiamo per ottenere la soluzione desiderata dalla soluzione del caso semplice
- ↗ **Integrazione:** riscriviamo la funzione integrando i vari casi (e' possibile averne piu' di uno)

PROGRAMMAZIONE RICORSIVA

[4]

○ ESEMPIO: LUNGHEZZA DI UNA LISTA

- ↗ **Specifica:** sia L una lista di N elementi con $N \geq 0$ ($N=0$ con lista vuota), $\text{lung}(L)=N$
- ↗ **Base:** se la lista è vuota ($L=NULL$) $N=0$
- ↗ **Ipotesi ricorsiva:** supponiamo che la funzione lung operi correttamente su di una lista piu' semplice, ad esempio la coda: $\text{lung}(L \rightarrow \text{Next})=N-1$
- ↗ **Passo:** come operiamo per avere la soluzione completa a partire dalla soluzione parziale $N=(N-1)+1$
- ↗ **Integrazione:**

```
int lung(lista_pt L)
{
    if (L==NULL)
        return 0;
    else
        return (1+lung(L->Next));
}
```

PROGRAMMAZIONE RICORSIVA

[5]

○ ESEMPIO:

↵ **Specifica:** data una lista di punti L ed un punto P restituisce 1 se trova P in L altrimenti 0

ITERATIVA

```
int trova(lista_pt L, punto P)
{
    int trovato=-1;
    while(trovato==-1)
    {
        if(L->Info==P)
            trovato=1;
        else if (L->Next==NULL)
            trovato=0;
        else
            L=L->Next
    }
    return trovato;
}
```

RICORSIVA

```
int trova(lista_pt L, punto P)
{
    if (L->Info==P)
        return 1;
    else if (L->Next==NULL)
        return 0;
    else
        return(trova(L->Next,P));
}
```