

Il concetto di ricorsione - elementi chiave
=====

Quando è utilizzabile.

Dato un problema P di tipo T, esso si presta ad essere risolto ricorsivamente se soddisfa le seguenti condizioni:

- 1) La soluzione del problema P è esprimibile in funzione della soluzione di un problema P' che:
 - a) è dello stesso tipo T di P;
 - b) ha "dimensione inferiore" rispetto a P.
- 2) Esiste una "dimensione elementare" per i problemi di tipo T, alla quale corrisponde un algoritmo di soluzione semplice.

Il concetto di "dimensione" va definito in relazione allo specifico tipo T di problema considerato, il che non è necessariamente possibile. Ad esempio in problemi che coinvolgono strutture dati a più elementi la "dimensione" può essere il numero di elementi della struttura. Per poter risolvere ricorsivamente i problemi di tipo T è poi necessario che i concetti di "dimensione inferiore" e "dimensione elementare" siano a loro volta definibili.

Esempi di problemi risolubili ricorsivamente:

- smontare una macchina complessa (prima separarla in sezioni, poi gestire lo smontaggio di ciascuna sezione);
- passare l'aspirapolvere in un appartamento (prima dividere l'area in sotto-aree fino a raggiungere una dimensione minima -ad esempio pari all'area che può essere pulita senza spostare i piedi-, poi gestire l'aspirazione di ciascuna sotto-area).

Elementi di un algoritmo ricorsivo per la risoluzione dei problemi di tipo T:

- 1) Una funzione per la risoluzione dei problemi di tipo T che:
 - se il problema ha complessità elementare (overosia se si è in uno dei casi elementari, che possono essere più d'uno), lo risolve;
 - altrimenti esprime la soluzione del problema in termini di soluzione di

un problema di tipo T di complessità inferiore, poi utilizza una chiamata a sé stessa per ottenere la soluzione del problema semplificato.

2) Una "condizione di risolubilità" che consente di stabilire se un problema di tipo T ha complessità elementare.

Vantaggi degli algoritmi ricorsivi.

Possono apparire quasi "magici": al primo impatto non è evidente "chi svolge effettivamente il lavoro"...

Il programmatore si limita a definire la soluzione del caso "elementare" ed il meccanismo di scomposizione, e lascia al processo di ricorsione stesso la gestione dei casi più complessi. Dunque il lavoro di soluzione del problema lo fa essenzialmente il meccanismo di scomposizione.

In pratica, risolvere un problema con la ricorsione significa eseguire la risoluzione di un problema complesso non tramite un algoritmo complesso, bensì tramite un meccanismo composto da un algoritmo per la riduzione della complessità del problema e di un algoritmo per la risoluzione del problema avente complessità minima.

Molti problemi ammettono sia una soluzione iterativa che una ricorsiva.

Quale sia quella ottimale dipende dai casi.

Tipicamente (per i problemi che la ammettono) la soluzione ricorsiva risulta più concisa di una iterativa, ma più complessa da capire.

Svantaggi degli algoritmi ricorsivi.

Lo svantaggio principale è costituito dal consumo di memoria. Ogni nuova chiamata della funzione ricorsiva aggiunge il relativo record di attivazione allo stack. Fino a quando l'ultima funzione chiamata da questa catena di chiamate ricorsive termina la sua esecuzione, tutti questi record di attivazione rimangono nello stack, utilizzando una quantità di memoria che dipende dalle variabili e dai parametri della funzione ma che può essere ingente. Peggio ancora, questa quantità di memoria è difficile da stimare in anticipo perché dipende dalle specifiche condizioni di funzionamento del programma in esecuzione, che influenzano il numero di chiamate ricorsive effettuate.

Il caso peggiore si ha quando le variabili vengono passate per valore alla funzione ricorsiva, giacché in tal caso viene creata una copia dei dati ogni

volta che avviene una nuova chiamata.

Un possibile svantaggio ulteriore si verifica quando la ricorsione viene usata nella soluzione di problemi per i quali essa non si presta bene (vedere in proposito il seguito di questo documento). In questi casi è facile che il programma risulti poco comprensibile.

Quando conviene usare la ricorsione?

Alcuni problemi si prestano per loro stessa natura ad una soluzione di tipo iterativo, mentre altri problemi sono più adatti ad una soluzione ricorsiva. Esistono anche problemi per i quali è possibile realizzare facilmente soluzioni sia di un tipo che dell'altro. Il modo migliore per acquisire l'esperienza necessaria a fare di volta in volta la scelta migliore è provare, quando si risolvono esercizi, a realizzare ciascuna funzione sia in modo iterativo che ricorsivo.