



Algoritmi



Sommario

- Definizione di informatica.
- Definizione di un calcolatore come esecutore.
- Gli algoritmi.



Informatica

- Nome

- Informatica=informazione+automatica.

- Definizione

- Scienza che si occupa dell'automazione del trattamento, modellazione, elaborazione delle informazioni.



Introduzione

- Un tipo di azioni compiute dall'uomo: soluzione di problemi.
- Un tipo di problema: elaborazione di informazione.
- Esempi:
 - calcolare l'area di un cerchio;
 - riconoscere il volto di una persona.



Elaborazione dell'informazione

- Un problema di elaborazione dell'informazione è caratterizzato da:
 - insieme di dati di partenza;
 - un risultato cercato;
 - una soluzione: una procedura che genera il risultato a partire dai dati di partenza.



La soluzione

- La conoscenza di **come** si risolve un problema e la **capacità** di risolverlo sono competenze distinte.
- Esempio: ognuno è capace di riconoscere un volto, ma come avviene questo riconoscimento? Come descrivere la procedura per riconoscere uno specifico volto?



La procedura di soluzione

- Può capitare di trovarsi di fronte ad un problema la cui soluzione debba essere attuata non da noi, ma da un altro soggetto.
- Il soggetto può non sapere come risolvere il problema, sebbene possa dichiarare la sua disponibilità ad attuare la soluzione nel momento in cui gli venisse insegnata.



La procedura di soluzione

- La procedura di soluzione deve allora essere realizzata in fasi distinte e successive:
 - analisi del problema e identificazione di una soluzione da parte del primo soggetto;
 - descrizione della soluzione da parte del primo soggetto in termini comprensibili al secondo soggetto;
 - interpretazione della soluzione da parte del secondo soggetto;
 - attuazione della soluzione da parte del secondo soggetto.



L'esecutore

- La procedura di soluzione deve essere descritta in una forma che l'esecutore sia in grado di interpretare in modo corretto.
- La soluzione deve specificare delle azioni che l'esecutore è in grado di attuare.
- Ogni esecutore è caratterizzato dalle sua capacità di interpretazione e di attuazione.



Il calcolatore

- I calcolatori sono degli esecutori di soluzioni che esseri umani hanno precedentemente identificato e descritto.
- I calcolatori hanno una notevole velocità di esecuzione e possono ripetere la stessa operazione producendo sempre lo stesso risultato un numero elevato di volte.



Il calcolatore

- Il calcolatore in quanto esecutore è caratterizzato da:
 - un linguaggio che è in grado di interpretare, con il quale devono essere descritte le soluzioni che vuole che esso attui;
 - istruzioni che è in grado di eseguire.



Problemi e algoritmi

- Se un problema è particolarmente semplice, l'esecutore potrebbe essere in grado di eseguire la soluzione direttamente.
- Esempio: determinare la superficie di un cerchio di raggio r .
- Ma se l'esecutore non conosce la formula risolutiva, la si deve indicare esplicitamente: $S = \pi r^2$.
- Tuttavia se l'esecutore non conosce come elevare un numero al quadrato, si ha che la soluzione contiene a sua volta un problema la cui soluzione deve essere descritta in modo esplicito.



Gerarchia di problemi

- In generale, per giungere alla descrizione della soluzione di un problema, si scompone il problema in sottoproblemi, e questi in sotto-sottoproblemi.
- Ci si ferma quando si giunge ad un **problema elementare** o **primitivo** la cui soluzione corrisponda ad una **azione elementare** che può essere direttamente compiuta dall'esecutore.
- Risolvere un problema equivale a risolvere una opportuna successione di problemi più semplici.



Soluzione effettiva

- Definiamo **effettiva** per un esecutore la soluzione di un problema quando:
 - l'esecutore è in grado di interpretare la descrizione di tale soluzione e associare ad essa le azioni che deve compiere per eseguirla;
 - l'esecutore è in grado di compiere tali azioni completandone l'esecuzione in tempo finito.



Soluzione mediante scomposizione

- La soluzione di un problema mediante la sua scomposizione in sottoproblemi è strutturata nel seguente modo:
 - se la soluzione del problema è effettiva, allora l'esecutore la attua;
 - altrimenti il problema viene scomposto in sottoproblemi, e per ognuno di questi si applica nuovamente la procedura.



Procedura effettiva

- L'insieme dei sottoproblemi viene risolto da una **procedura effettiva** quando:
 - tutti i problemi sono elementari;
 - è fissato l'ordine di soluzione dei problemi;
 - è specificato il modo in cui un problema utilizza i risultati dei problemi che lo precedono.



Problemi e procedure effettive

- Ad ogni scomposizione di un problema in sottoproblemi può essere associata una procedura effettiva quando vengano considerati come elementari tutti i sottoproblemi.
- I concetti di problema elementare e azione elementare sono strettamente associati ed evidenziano l'aspetto **descrittivo** e **esecutivo** delle procedure risolutive.



L'esecutore (raffinamento della definizione)

- Un esecutore è caratterizzato da:
 - il linguaggio che è in grado di comprendere;
 - l'insieme delle azioni che è in grado di compiere;
 - l'insieme delle regole che ad ogni costrutto linguistico sintatticamente corretto associano le rispettive azioni da compiere.



Ambiguità

- Finché la soluzione di un problema viene descritta in termini informali (come ad esempio tra gli esseri umani), può rimanere **l'ambiguità** circa l'attuabilità della soluzione da parte dell'esecutore (la sua effettività).
- Si ha ambiguità quando due soggetti giudicano come effettiva la stessa soluzione di un problema ma poi compiono azioni che producono **risultati differenti**.
- Per rimuovere tale ambiguità si deve **formalizzare** la definizione di un esecutore.



L'esecutore (formalizzazione della definizione)

- Caratterizzazione **sintattica** dell'esecutore: il linguaggio che l'esecutore è in grado di interpretare deve essere definito in termini formali.
- l'insieme delle azioni elementari che l'esecutore è in grado di compiere deve essere unicamente definito, e tali azioni devono essere deterministiche, cioè l'esecuzione di una stessa azione deve sempre produrre lo stesso risultato.
- Caratterizzazione **semantica** dell'esecutore: l'insieme delle regole di associazione tra costrutti del linguaggio e azioni deve essere univocamente definito.



Algoritmi e programmi

- Le soluzioni effettive per esecutori caratterizzati formalmente sono chiamate **algoritmi**.
- Quando l'esecutore è un calcolatore, gli algoritmi vengono detti **programmi**.
- Il linguaggio formale per la loro descrizione è detto **linguaggio di programmazione**.



Algoritmo

- Il termine *algoritmo* deriva dal nome del matematico arabo *al-Khuwarizmi* (IX secolo d.C.).
- Definizione: un algoritmo è una successione ordinata di *istruzioni* (o passi) che definiscono le operazioni da eseguire su dei dati per risolvere una classe di problemi.



Sviluppo di un programma

- Il processo di sviluppo di un programma è organizzato in:
 - analisi del problema e identificazione di una soluzione;
 - formalizzazione della soluzione e definizione dell'algoritmo risolutivo;
 - programmazione, cioè scrittura dell'algoritmo in un linguaggio di programmazione "di alto livello";
 - traduzione del programma in un "linguaggio macchina", direttamente interpretabile dalla macchina.



Linguaggi di alto livello e linguaggio macchina

- I linguaggi di alto livello sono più facilmente comprensibili dagli esseri umani, ma **sono** sempre linguaggi formali.
- Il linguaggio macchina è un linguaggio formale comprensibile direttamente da uno specifico calcolatore.
- La traduzione da quello di alto livello a quello macchina può essere fatta automaticamente in virtù delle proprietà formali di entrambi.



La macchina universale

- Un elaboratore o **computer** è una macchina *digitale, elettronica, automatica*, capace di effettuare trasformazioni o *elaborazioni* sui dati
 - **digitale**=l'informazione è rappresentata in forma numerica discreta;
 - **elettronica**=la logica di manipolazione e la memorizzazione sono implementate con tecnologie di tipo elettronico (piuttosto che di tipo meccanico);
 - **automatica**=è in grado di eseguire una successione di operazioni in modo autonomo (cioè senza intervento di un operatore umano).



La macchina universale

- Le operazioni sono scritte sotto forma di un *programma*.
- Il programma e i dati su cui deve operare sono registrati in un *dispositivo di memoria*.
- Un dispositivo detto *unità di controllo* legge il programma e lo esegue sui dati.
- Questo modo di operare è detto *architettura di Von Neumann*.



La macchina universale

- Il programma permette di risolvere un problema in funzione dei dati.
- Se i dati possono cambiare e il programma risolve sempre il problema, allora si dice che il programma risolve una classe di problemi.
- Esempio: l'algoritmo per la somma di due numeri funziona per qualsiasi coppia di numeri (è indipendente dai due numeri dati in ingresso).



La macchina universale

- L'elaboratore si dice universale perché può essere usato per risolvere qualsiasi problema la cui soluzione possa essere descritta con un programma.
- Per ogni classe di problemi è necessario fornire un programma adeguato.



Esempi

- Diamo un esempio di un algoritmo semplice e di un algoritmo che usi il risultato di un altro algoritmo per risolvere un problema più complesso.



Determinazione del maggiore di due numeri interi

- Occorre definire quali problemi sono elementari, cioè quali problemi hanno una soluzione che può essere eseguita direttamente senza dover ricorrere ad altre scomposizioni.
- Supponiamo che la differenza tra due interi e la valutazione del segno positivo o negativo di un numero siano problemi elementari.



Determinazione del maggiore di due numeri interi

- P0: inizio dell'algoritmo;
- P1: leggi un valore dall'esterno e inseriscilo nella variabile x ;
- P2: leggi un valore dall'esterno e inseriscilo nella variabile y ;
- P3: calcola la differenza $d \leftarrow x - y$;
- P4: se d ha segno positivo, vai al passo P5, altrimenti al passo P6;
- P5: stampa "il massimo è" e il valore di x ; vai al passo P7;
- P6: stampa "il massimo è" e il valore di y ; vai al passo P7;
- P7: fine dell'algoritmo.



Scomposizione in sottoproblemi

- Per problemi più complessi il numero di passi cresce notevolmente.
- Per semplificare la scrittura di un algoritmo lo si può scrivere in funzione di sottoproblemi non elementari purché di essi sia nota la scomposizione in problemi elementari.
- Questi problemi dalla soluzione nota sono detti **problemi terminali**.



Scomposizione in sottoproblemi

- L'esecuzione di un algoritmo può essere pensata in termini di soluzione per un insieme di problemi terminali.
- In un linguaggio di programmazione
 - alla soluzione dei problemi terminali elementari corrisponde il concetto di **istruzione**;
 - alla soluzione dei problemi terminali non elementari corrisponde il concetto di **sottoprogramma** (procedura o funzione).



Determinazione del maggiore di tre numeri interi

- Possiamo considerare terminale l'algoritmo per la soluzione del problema del massimo tra due numeri interi.
- Il problema può dunque essere scomposto come segue:
 - P1: se x è maggiore di y , allora esegui P2, altrimenti esegui P3;
 - P2: la soluzione è il maggiore tra x e z ;
 - P3: la soluzione è il maggiore tra y e z .



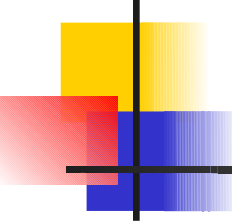
Determinazione del maggiore di n numeri interi

- Si può generalizzare il procedimento come segue:
 - P1: trova il maggiore tra i primi due numeri;
 - P2: trova il maggiore tra il terzo ed il risultato del passo precedente;
 - P3: trova il maggiore tra il quarto ed il risultato del passo precedente;
 -



Determinazione del maggiore di n numeri interi

- Più elegantemente:
 - P1: trova il maggiore tra i primi due numeri;
 - P2: finché ci sono numeri esegui P3, altrimenti P4;
 - P3: trova il maggiore tra il nuovo numero e quello trovato al passo precedente;
 - P4: la soluzione è l'ultimo numero trovato al passo P3.



Nota

- Il passo P2 mostra una struttura usata spesso nella descrizione dei problemi ripetitivi: “finché condizione, ripeti azione”.
- Tale struttura indica che l’azione deve essere eseguita ripetutamente valutando ogni volta la condizione.
- In questo modo si ottiene una formulazione molto concisa e indipendente da ogni specifico valore di n .
- Un problema che ammette una soluzione di questo tipo si dice che ha una soluzione di tipo **iterativo**.



Nota

- I linguaggi di programmazione hanno modi molto compatti per esprimere diverse strutture iterative per controllare il flusso dell'elaborazione.



Gioco dell'11

- Ci sono 11 oggetti e due giocatori, A e B. I giocatori a turno prendono fino a un massimo di 3 oggetti. Perde chi prende l'ultimo oggetto. Problema: c'è un algoritmo che permette al primo giocatore A di vincere sempre? Se sì, quale?



Gioco dell'11

- Algoritmo:

- 1. Inizio dell'algoritmo.
- 2. A preleva 2 oggetti.
- 3. B preleva k oggetti.
- 4. A preleva $4-k$ oggetti.
- 5. Se gli oggetti non sono finiti, allora ritorna all'istruzione 3.
- 6. Fine dell'algoritmo.