

Processo di risoluzione di un problema ingegneristico

1. Capire l'essenza del problema.
2. Raccogliere le informazioni disponibili. Alcune potrebbero essere disponibili in un secondo momento.
3. Determinare le informazioni da trovare.
4. Semplificare il problema quanto basta per consentire di ricavare le informazioni richieste. Definire tutte le ipotesi del problema.
5. Disegnare uno schema e assegnare i nomi alle variabili .
6. Determinare i principi fondamentali da applicare.
7. Esaminare il metodo proposto in generale e considerare altri approcci prima di passare ai dettagli.

1
LEZ 03 - ALGORITMI

Processo di risoluzione di un problema ingegneristico

8. Assegnare un nome ad ogni passo del processo di risoluzione.
9. Se il problema viene risolto con un programma, controllare i risultati utilizzando una versione semplice del problema.
Verificare le dimensioni e le unità di misura (controllo dimensionale) e stampare i risultati intermedi del calcolo, per scoprire eventuali errori nascosti.

2
LEZ 03 - ALGORITMI

Processo di risoluzione di un problema ingegneristico

10. Effettuare una verifica pratica della soluzione ottenuta. Ha senso il risultato? Stimare l'intervallo dei risultati e confrontarlo con quello ottenuto. La soluzione non dovrà avere una precisione maggiore di quella che possa essere giustificata da:
 - (a) La precisione delle informazioni note
 - (b) Le ipotesi semplificative.
 - (c) I requisiti del problema.Interpretare i risultati matematici. Se si ottengono più soluzioni, non scartare una soluzione se prima non è chiaro il suo significato. Le formule matematiche potrebbero segnalare qualche cosa di nuovo, offrendo la possibilità di capire meglio il problema.

3
LEZ 03 - ALGORITMI

Soluzione al computer

1. Definire i termini del problema in modo conciso.
2. Specificare i dati che dovranno essere utilizzati dal programma. Questo è l' "input."
3. Specificare i dati che dovranno essere generati dal programma. Questo è l' "output."
4. Elaborare a mano o a linea di comando i vari passaggi per ottenere la soluzione. Utilizzare un insieme di dati semplici.

4
LEZ 03 - ALGORITMI

Soluzione al computer

5. Scrivere ed eseguire il programma.
6. Verificare il risultato fornito dal programma con quello ottenuto "a mano".
7. Eseguire il programma con i dati di input ed effettuare una verifica pratica dell'output.
8. Se si vuole riutilizzare il programma come strumento generico, testarlo con una serie appropriata di dati; effettuare una verifica pratica dei risultati.

INTRODUZIONE AGLI ALGORITMI

- Prima di riuscire a scrivere un programma, abbiamo bisogno di conoscere un metodo risolutivo, cioè un metodo che a partire dai dati di ingresso fornisce i risultati attesi.
- Se voglio calcolare una moltiplicazione, posso usare diversi metodi:

- mi baso sull'addizione:

$$13 \times 12 = 13 + 13 + 13 + \dots + 13$$

- oppure faccio il calcolo in colonna

$$\begin{array}{r} 13 \times \\ 12 = \\ \hline 26 + \\ 13 = \\ \hline 156 \end{array}$$

INTRODUZIONE AGLI ALGORITMI

- Se dobbiamo spiegare ad una persona come fare una moltiplicazione, possiamo fare loro un esempio:

$$13 \times 12 = 13 + 13 + 13 + \dots + 13$$

- però così abbiamo spiegato solo come si fa una particolare moltiplicazione: 13×12 .
- Vogliamo spiegare un metodo che valga "sempre", per tutti i numeri. Dobbiamo innanzitutto definire **per quali valori di ingresso** funziona il nostro metodo

INTRODUZIONE AGLI ALGORITMI

- La persona che ci ascolta è in grado di imparare da esempi. Il calcolatore no
 - non possiamo spiegarlo con un esempio, dobbiamo dire quali passi deve svolgere
- Proviamo così:

$$A \times B = A + A + A + \dots + A$$

Cosa vogliono dire i puntini?
Ambiguo!

INTRODUZIONE AGLI ALGORITMI

- La persona che ci ascolta è in grado di imparare da esempi. Il calcolatore no
 - non possiamo spiegarlo con un esempio, dobbiamo dire quali passi deve svolgere
- Proviamo così:

$$A \times B = \underbrace{A + A + A + \dots + A}_{B \text{ volte}}$$

- *Ma cosa vuol dire una somma con B termini?*
- *Es, cosa vuol dire 0 termini?*
- *Quale istruzione deve essere ripetuta B volte?*

9
LEZ 03 - ALGORITMI

INTRODUZIONE AGLI ALGORITMI

- Se vogliamo scrivere un libro di matematica di base, spiego come si fa una moltiplicazione. Devo scriverlo in modo che chiunque capisca il metodo
- non deve essere ambiguo
- dobbiamo dire quali sono i prerequisiti, le istruzioni di base che l'esecutore deve saper compiere
 - per capire questo algoritmo bisogna sapere come si calcola un'addizione
- dobbiamo dire su quali valori di ingresso si può applicare il metodo
 - sui naturali

10
LEZ 03 - ALGORITMI

SOLUZIONE: CALCOLO DEL PRODOTTO

IL PIU' PICCOLO NUMERO REALE >0

1. Assegna a R il valore 1
 2. Dividi R per 2 e metti il risultato in R
 3. Torna al passo 2
 4. Stampa R
- Non stampa mai il valore R
 - Non termina

11
LEZ 03 - ALGORITMI

12
LEZ 03 - ALGORITMI

RISOLUZIONE DI PROBLEMI

- La risoluzione di un problema è il processo che, dato un problema e individuato un opportuno metodo risolutivo, trasforma i dati iniziali nei corrispondenti risultati finali.
- Affinché la risoluzione di un problema possa essere realizzata attraverso l'uso del calcolatore, tale processo deve poter essere definito come *sequenza di azioni elementari*.

ALGORITMO

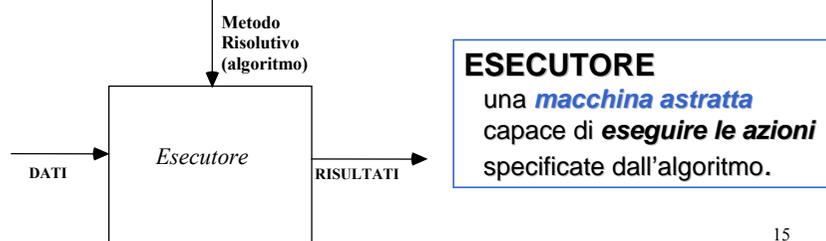
- Il termine *Algoritmo* deriva dal nome del matematico persiano

Muhammad ibn Mūsā al-Khwārizmī
(ca. 780-850)



ALGORITMO

- Un algoritmo è una sequenza **finita** di mosse che risolve **in un tempo finito** una *classe* di problemi.
- L'esecuzione delle azioni *nell'ordine specificato dall'algoritmo* consente di ottenere, a partire dai dati di ingresso, i risultati che risolvono il problema



ALGORITMI: PROPRIETÀ

- **Eseguibilità**: ogni azione dev'essere eseguibile dall'esecutore *in un tempo finito*
- **Non-ambiguità**: ogni azione deve essere *univocamente interpretabile* dall'esecutore.
L'italiano è ambiguo, come si vede bene dagli indovinelli
IL MESE DI MAGGIO:
"Ratto trascorre e a noi rose dispensa"
- **Finitezza**: il numero totale di azioni da eseguire, per ogni insieme di dati di ingresso, deve essere finito

ALGORITMI: PROPRIETÀ (2)

Quindi, l'algoritmo deve:

- essere *applicabile a qualsiasi insieme di dati di ingresso* appartenenti al **dominio di definizione** dell'algoritmo
- essere costituito da operazioni appartenenti ad un determinato **insieme di operazioni fondamentali**
- essere costituito da **regole non ambigue**, cioè interpretabili in modo **univoco** qualunque sia l'esecutore (persona o "macchina") che le legge

17
LEZ 03 - ALGORITMI

UNA VOLTA DECISO L'ALGORITMO

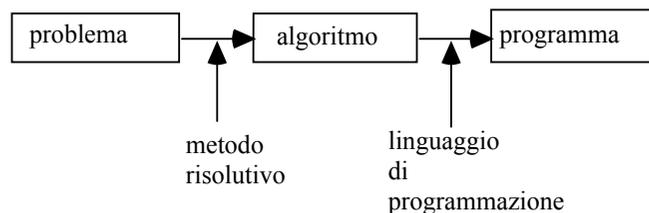
- Una volta che ho deciso l'algoritmo, devo fare in modo che l'**elaboratore** sia in grado di eseguirlo
- Le "*mosse elementari*" devono essere eseguibili dal calcolatore (quindi devo sapere quali **istruzioni** il calcolatore può eseguire)
- Le istruzioni vengono eseguite sui **dati** e forniscono dei **risultati**
- L'algoritmo deve essere scritto in maniera formale: codificato in un preciso **linguaggio di programmazione**

18
LEZ 03 - ALGORITMI

ALGORITMO & PROGRAMMA

Passi per la risoluzione di un problema:

- individuazione di un procedimento risolutivo
- scomposizione del procedimento in un insieme ordinato di azioni ➡ **ALGORITMO**
- rappresentazione dei dati e dell'algoritmo attraverso un formalismo comprensibile dal calcolatore ➡ **LINGUAGGIO DI PROGRAMMAZIONE**



19
LEZ 03 - ALGORITMI

PROGRAMMA

Un **programma** è un **testo** scritto in accordo alla **sintassi** e alla **semantica** di un linguaggio di programmazione.

Un **programma** è la **formulazione testuale**, in un certo linguaggio di programmazione, di un **algoritmo** che risolve un dato *problema*.

20
LEZ 03 - ALGORITMI

UN ESEMPIO DI PROGRAMMA (in linguaggio C)

```
main() {
    int A, B;
    printf("Immettere due numeri: ");
    scanf("%d %d", &A, &B);
    printf("Somma: %d\n", A+B);
}
```

ALCUNE DOMANDE FONDAMENTALI

- Quali istruzioni esegue un elaboratore?
- Quali problemi può risolvere un elaboratore?
- *Esistono problemi che un elaboratore non può risolvere?*
- Che ruolo ha il linguaggio di programmazione?

PROBLEMI DA RISOLVERE

- I problemi che siamo interessati a risolvere con l'elaboratore sono di natura molto varia.
 - *Dati due numeri trovare il maggiore*
 - *Dato un elenco di nomi e relativi numeri di telefono trovare il numero di telefono di una determinata persona*
 - *Dati a e b, risolvere l'equazione $ax+b=0$*
 - *Stabilire se una parola viene alfabeticamente prima di un'altra*
 - *Somma di due numeri interi*
 - *Ordinare una lista di elementi*
 - *Calcolare il massimo comun divisore fra due numeri dati.*
 - *Calcolare il massimo in un insieme.*

RISOLUZIONE DI PROBLEMI

- La descrizione del problema non fornisce (in generale) un metodo per risolverlo.
 - Affinché un problema sia risolvibile è però necessario che la sua definizione sia chiara e completa
- Non tutti i problemi sono risolvibili attraverso l'uso del calcolatore. Esistono classi di problemi per le quali la soluzione automatica non è proponibile. Ad esempio:
 - se il problema presenta infinite soluzioni
 - per alcuni dei problemi **non è stato trovato** un metodo risolutivo
 - per alcuni problemi è stato dimostrato che **non esiste** un metodo risolutivo automatizzabile

ALGORITMI: ESEMPI

- **Soluzione dell'equazione $ax+b=0$**
 - leggi i valori di a e b
 - calcola $-b$
 - dividi quello che hai ottenuto per a e chiama x il risultato
 - stampa x

NOTA: per denotare dati nell'algoritmo si utilizzano
VARIABILI ossia nomi simbolici

ALGORITMI: ESEMPI

- **Calcolo del massimo di una sequenza di numeri $a_1 \dots a_n$**
 - 1 livello di specifica
 - Scegli il primo elemento come massimo provvisorio $max \leftarrow a_1$
 - Per ogni elemento a_i dell'insieme: se $a_i > max$ eleggi a_i come nuovo massimo provvisorio: $max \leftarrow a_i$
 - Il risultato è max

ALGORITMI: ESEMPI

- **Stabilire se una parola P viene alfabeticamente prima di una parola Q**
 - **leggi** P, Q
 - **ripeti quanto segue:**
 - **se** prima lettera di $P <$ prima lettera di Q
 - **allora** scrivi vero
 - **altrimenti se** prima lettera $P > Q$
 - **allora** scrivi falso
 - **altrimenti** (le lettere sono =)
 - toglì da P e Q la prima lettera
 - **fino** a quando hai trovato le prime lettere diverse.

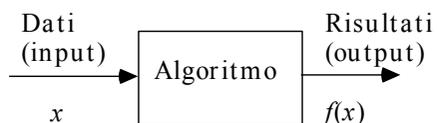
ALGORITMI EQUIVALENTI

- Due algoritmi si dicono **equivalenti** quando:
- hanno lo stesso **dominio di ingresso**;
 - hanno lo stesso **dominio di uscita**;
 - in corrispondenza degli **stessi valori del dominio di ingresso producono gli stessi valori nel dominio di uscita**.

ALGORITMI EQUIVALENTI (2)

Due algoritmi *equivalenti*

- forniscono lo **stesso risultato**
- ma possono avere ***diversa efficienza***
- e possono essere **profondamente diversi !**



ALGORITMI EQUIVALENTI (3)

ESEMPIO: calcolo del M.C.D. fra due interi M, N

- **Algoritmo 1**
 - Calcola l'insieme A dei divisori di M
 - Calcola l'insieme B dei divisori di N
 - Calcola l'insieme C dei divisori comuni = $A \cap B$
 - Il risultato è il massimo dell'insieme C
- **Algoritmo 2 (di Euclide)**

$$\text{MCD}(M,N) = \begin{cases} M \text{ (oppure } N) & \text{se } M=N \\ \text{MCD}(M-N, N) & \text{se } M>N \\ \text{MCD}(M, N-M) & \text{se } M<N \end{cases}$$

ALGORITMI EQUIVALENTI (4)

ESEMPIO: calcolo del M.C.D. fra due interi M, N

- **Algoritmo 2 (di Euclide)**
 - Finché $M \neq N$:
 - se $M > N$, sostituisci a M il valore $M' = M - N$
 - altrimenti sostituisci a N il valore $N' = N - M$
 - Il Massimo Comun Divisore è il valore finale ottenuto quando M e N diventano uguali

$$\text{MCD}(M,N) = \begin{cases} M \text{ (oppure } N) & \text{se } M=N \\ \text{MCD}(M-N, N) & \text{se } M>N \\ \text{MCD}(M, N-M) & \text{se } M<N \end{cases}$$

ALGORITMI EQUIVALENTI (5)

**Gli algoritmi 1 e 2 sono equivalenti...
...ma hanno efficienza ben diversa!!**