Programmazione

Dipartimento di Matematica

Ing. Cristiano Gregnanin

Corso di laurea in Matematica

29 febbraio 2016

INTRODUZIONE AGLI ALGORITMI

Prima di riuscire a scrivere un programma, bisogna conoscere un metodo risolutivo, cioè un metodo che a partire dai dati ingresso fornisce i risultati attesi.
Per calcolare una moltiplicazione si possono usare diversi metodi:

- addizione $13x12 = 13 + 13 + ... + 13$
- calcolo di $13x12$ in colonna.

2 strade per giungere al medesimo risultato

INTRODUZIONE AGLI ALGORITMI

Attraverso un esempio è possibile spiegare come fare una moltiplicazione, però si è spiegato solamente come fare una moltiplicazione particolare e non tutte le moltiplicazioni. Per spiegare un metodo che valga per tutti i numeri occorre definire per quali **valori di ingresso** funziona questo metodo

INTRODUZIONE AGLI ALGORITMI

Si potrebbe cercare di dire al calcolatore di eseguire $A + A + A + ... + A$ per B volte.
Problematiche:

- Cosa vuol dire una somma con B termini?
- Cosa vuol dire 0 termini?
- Quale istruzione deve essere ripetuta B volte?

INTRODUZIONE AGLI ALGORITMI

Per codificare un meccanismo di risoluzione di un problema bisogna adottare accorgimenti ben precisi:

- non deve essere ambiguo
- occorre esplicitare i prerequisiti ovvero le istruzioni di base che che l'esecutore deve saper compiere. *Ad esempio per capire questo algoritmo bisogna sapere come si calcola un'addizione*
- occorre esplicitare su quali valori di ingresso si può applicare il metodo. *Ad esempio i numeri naturali.*

INTRODUZIONE AGLI ALGORITMI

Esempio: calcolo del più piccolo numero reale > 0

1. Assegna ad R il valore 1
2. dividi R per 2 e metti il risultato in R
3. vai al passo 2
4. stampa R

INTRODUZIONE AGLI ALGORITMI

Esempio: calcolo di Pigreco

Formula di Leibniz per PI:

$$\frac{\Pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + ... \tag{1}$$

1. Assegna a P il valore 0
2. Assegna a S il valore 1
3. Assegna a D il valore 1
4. Calcola $\frac{S}{D}$, sommalo a P e metti il risultato in P
5. Cambia segno a S
6. Somma 2 a D e metti il risultato in D
7. Vai al passo 4
8. Moltiplica P per 4
9. Stampa P

RISOLUZIONE DEI PROBLEMI

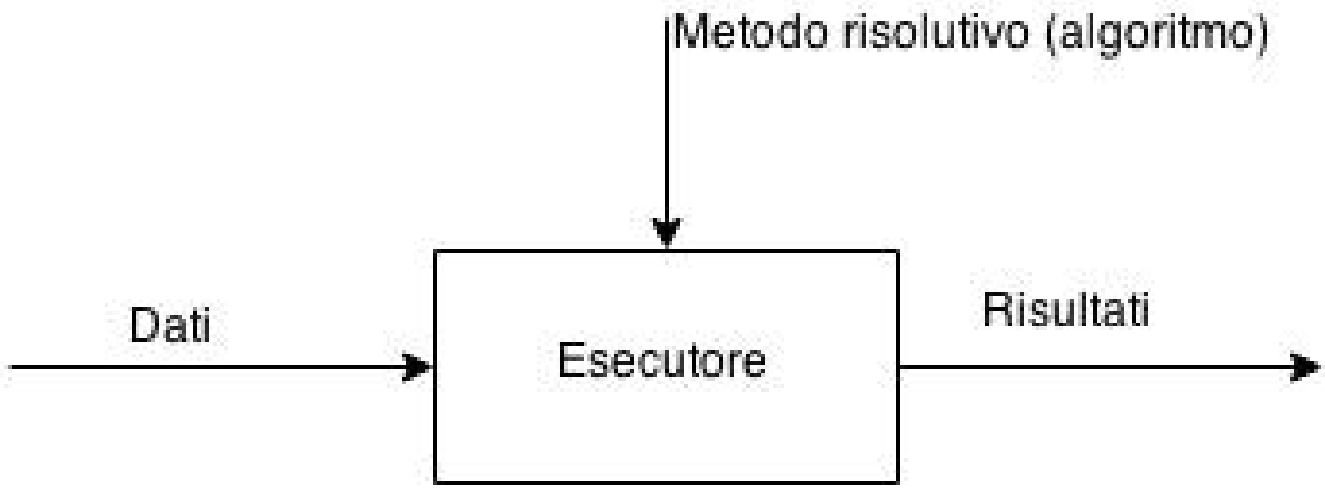
La risoluzione di un problema è un processo che, dato un problema ed individuato un opportuno metodo risolutivo, trasforma i dati iniziali nei corrispondenti dati finali.
Affinché la risoluzione di un problema possa essere realizzata attraverso l'uso del calcolatore, tale processo deve poter essere definito come **sequenza di azioni elementari.**
Il termine algoritmo deriva dal nome del matematico persiano **Muhammad ibn musa al-Khwarizmi.**

ALGORITMO

Definizione

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.

L'esecutore è una macchina astratta capace di eseguire le azioni specificate nell'algoritmo



ALGORITMO

Proprietà

- **Eseguibilità:** ogni azione deve essere eseguibile dall'esecutore in un tempo finito
- **Non ambiguità:** ogni azione deve essere univocamente interpretabile dall'esecutore
- **Finitezza:** il numero totale delle azioni da eseguire, per ogni insieme di dati di ingresso, deve essere finito.

ALGORITMO

Proprietà

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

ALGORITMO

Dopo aver deciso l'algoritmo

- Occorre 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**

ALGORITMO e 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**)

Si può affermare che: **Quando per un problema si definisce un metodo risolutivo (avente le proprietà viste in precedenza) si sta generando un algoritmo. In seguito, quando l'algoritmo viene codificato attraverso un linguaggio di programmazione allora si genera un programma. Il programma(reale) attua l'algoritmo(astratto)**

ALGORITMO e PROGRAMMA

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** di un **algoritmo** che risolve un dato problema

ESEMPIO DI UN PROGRAMMA

```
# This program adds two numbers
# Store input numbers

num1 = input('Enter first number: ')
num2 = input('Enter second number: ')

# Add two numbers
sum = float(num1) + float(num2)

# Display the sum
print(sum)
```

ELABORATORE ELETTRONICO

- Il calcolatore elettronico è uno strumento in grado di eseguire insiemi di **azioni elementari**
- Le azioni vengono *eseguite* su oggetti (**dati**) per *produrre* altri oggetti (**risultati**)
- L'esecuzione di azioni viene richiesta all'elaboratore attraverso *frasi* scritte in qualche *linguaggio*

PROGRAMMAZIONE

L'attività con cui si predispone l'elaboratore a **eseguire un particolare insieme di azioni su particolari dati**, allo scopo di *risolvere un problema*

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. Esempi:

- Dati due numeri trovare il maggiore
- Dato un elenco di nomi e relativi numeri di telefono trovare il numero di telefono di una persona
- Dati a e b, risolvere $ax + b = 0$
- Stabilire se una parola viene alfabeticamente prima di un'altra
- Somma di 2 numeri interi
- Ordinare una lista di elementi
- Calcolare il massimo comun divisore fra 2 numeri reali
- Calcolare il massimo in un insieme

RISOLUZIONE DEI 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

RISOLUZIONE DEI PROBLEMI

Pare essere ragionevole concentrarsi su problemi che ammettono un metodo risolutivo, ovvero su **funzioni calcolabili.**
Tecnologie e metodi di programmazione:

- **Tecnologie:** strumenti per lo sviluppo di programmi
- **Metodologie:** metodi per l'utilizzo corretto ed efficace delle tecnologie di programmazione.

ALGORITMI E PROGRAMMI

- Ogni elaboratore è una macchina in grado di eseguire azioni elementari su oggetti detti **dati**
- L'esecuzione delle azioni è richiesta all'elaboratore tramite comandi elementari chiamati **istruzioni** espresse attraverso un opportuno formalismo: **il linguaggio di programmazione**
- La formulazione testuale di un algoritmo in un linguaggio comprensibile a un elaboratore è detta **programma**

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

Osservazione: Per denotare dati nell'algoritmo si utilizzano variabili ossia nomi simbolici

ALGORITMI: ESEMPI

Calcolo del massimo di una sequenza di numeri $a_1 .... a_n$

- Scegli il primo elemento come massimo provvisorio: $max = a_1$
- **Per ogni** elemento a_i dell'insieme valuta: **se** $a_i > max$ **allora** eleggi a_i come nuovo massimo provvisorio $max = a_i$
- Il risultato è max

ALGORITMI: ESEMPI

Stabilire se una parola P viene alfabeticamente prima di una parola Q

- **Leggi P,Q**
- **Ripeti quanto segue:**
  1. **se** prima lettera di P $<$ prima lettera di Q
  2. **allora** scrivi *vero*
  3. **altrimenti** se prima lettera di P $>$ Q
  4. **allora** scrivi *falso*
  5. **altrimenti**
  6. togli da P e Q la prima lettera
- **fino** a quando hai trovato le prime lettere diverse

ALGORITMI: ESEMPI

Somma degli elementi dispari in un insieme: Detto INS l'insieme di elementi considero un elemento X di INS alla volta senza ripetizioni. Se X è dispari, sommo X a un valore S inizialmente posto uguale a 0. Se X è pari non compio alcuna azione.
Somma di due numeri X e Y Si supponga di avere a disposizione come mossa elementare solo l'incremento e non la somma di interi.

ALGORITMI: ESEMPI

Soluzione: *Somma di due numeri X e Y.*
Incrementare il valore di Z, inizialmente posto uguale a X per Y volte. Ovvero:

- poni $Z = X$
- poni $U = 0$
- finché U è diverso da Y
  1. incrementa Z ($Z = Z + 1$)
  2. incrementa U ($U = U + 1$)
- il risultato è Z

ALGORITMI EQUIVALENTI

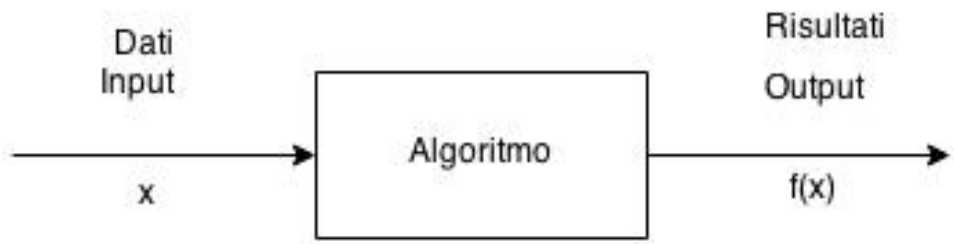
Due algoritmi si dicono **equivalenti** quando:

- 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

Due algoritmi **equivalenti**

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



ALGORITMI EQUIVALENTI

Esempio: calcolo del MCD fra due interi M,N

- algoritmo 1
  1. Calcola l'insieme A dei divisori di M
  2. Calcola l'insieme B dei divisori di N
  3. Calcola l'insieme C dei divisori comuni
  4. Il risultato è il massimo dell'insieme C
- algoritmo 2 (di Euclide)
  1. **Se** $M = N$ allora M (oppure N)
  2. **Se** $M > N$ allora MCD(M-N,N)
  3. **Se** $M < N$ allora MCD(M,N-M)

ALGORITMI EQUIVALENTI

Algoritmo 2 (di Euclide)

- Finché M != N
  1. **Se** $M > N$ allora sostituisci a M il valore M-N
  2. **altrimenti** sostituisci a N il valore N-M
- Il MCD è il valore finale ottenuto quando M e N diventano uguali

ALGORITMI EQUIVALENTI

Gli algoritmi 1 e 2 sono equivalenti... Ma hanno efficienza ben diversa. Perchè?
Esempio: provare a calcolare l'MCD di 324543324 e 654345432