

Algoritmi = Dati + Istruzioni

Fondamenti di Informatica A

Percorso di Preparazione agli Studi di Ingegneria

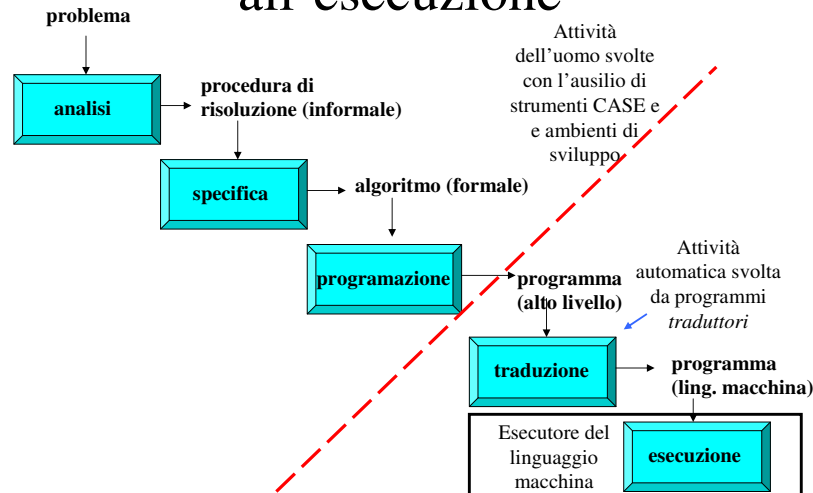
Università degli Studi di Brescia

Dott.ssa Daniela Fogli

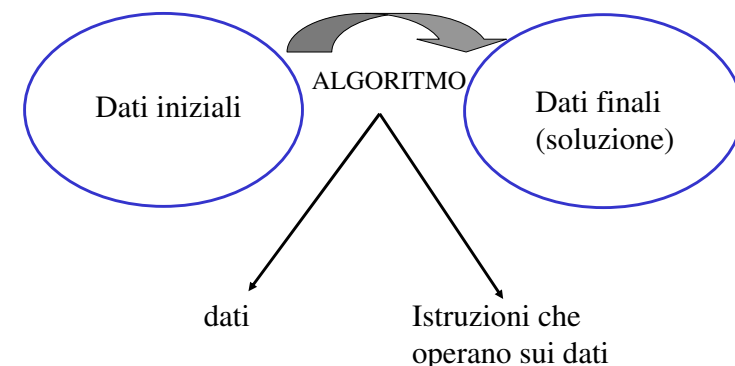
Dal problema a un programma comprensibile dal calcolatore

1. **Analisi** del problema e **identificazione** procedura di risoluzione
2. Specifica dell'**algoritmo** risolutivo
3. **Programmazione**
4. **Traduzione** del programma in linguaggio macchina
5. **Esecuzione** del programma

Dall'analisi del problema all'esecuzione



Algoritmo = dati + istruzioni



Dati e istruzioni

- **Dati:**
 - Numeri (naturali, interi, reali, ...)
 - Caratteri alfanumerici (a, b, c, ...)
 - Dati logici (vero, falso)
 - Vettori di elementi, matrici, ... ([1,2,3], [[1,1],[1,2],...])
- **Istruzioni:**
 - Istruzioni di **ingresso/uscita** (es. *leggi, visualizza,...*)
 - Istruzioni **aritmetico-logiche** (es. $c = a * b$)
 - Istruzioni di **controllo** (es. *se ... allora, ripeti...*)

Come posso classificare i dati?

- In base alla **visibilità** da parte dell'utente
 - Visibile (di ingresso o uscita)
 - Trasparente (dati temporanei di supporto)
- In base alla **variabilità** nel tempo
 - Costanti
 - Variabili (acquisizione dall'esterno o assegnazione)
- In base alla **struttura**
 - Elementari (interi, alfanumerici, logici, ...)
 - Strutturati (vettori, matrici, ...)

Variabili, espressioni e assegnamenti

- **Variabile:** “contenitore” di dati
- Ha un **nome** (es. x)... nel calcolatore i nomi delle variabili identificano una ben precisa locazione di memoria
- Ad una variabile può essere assegnato un **valore**:
es. $x \leftarrow 10$... nel calcolatore diventa il contenuto della locazione di memoria identificata dal nome della variabile
- Le variabili possono comparire in **espressioni aritmetiche** (es. $x-y$) o **logiche**, e in **predicati logici**
- Le espressioni possono essere assegnate ad altre variabili: ad es. scriveremo $d \leftarrow x-y$ per indicare che d conterrà il risultato dell'espressione
- Si potrebbe anche usare x o y al posto di d , in questo modo si sovrascrive il precedente valore

Espressioni aritmetiche

... sono formate da:

Operandi: variabili e costanti (si ipotizza di considerare solo numeri interi)

Operatori: addizione (+), sottrazione (-), moltiplicazione (*), divisione intera (/), modulo (mod)

Semantica: quella usuale dell'aritmetica

Esempi

$$(a \bmod 5) + b * 3$$

$$(b - c)/2 + 3 * a$$

Espressioni logiche

... sono formate da:

Operandi: espressioni aritmetiche

Operatori: **operatori relazionali** di uguaglianza (=), minoranza (<) e maggioranza (>) fra numeri

Semantica: quella delle disequazioni fra numeri

Esempi

$$a = b$$

$$(b + d) < (10 * c)$$



Predicati logici

... sono formati da:

Operandi: espressioni logiche e/o predicati

Operatori: **operatori logici** di congiunzione (AND), disgiunzione (OR) e negazione (NOT)

Semantica: quella dell'algebra di Boole

Esempi

$$(a < b) \text{ AND } (b < c) \text{ OR } (c = d+2)$$

$$\text{Not}((a=10*b) \text{ AND } (b = c)) \text{ OR } (c = 20)$$



Rappresentazione degli algoritmi

1. Linguaggio naturale/pseudo-codice
2. Diagrammi di flusso
3. Linguaggi di programmazione

Descrizione in linguaggio naturale

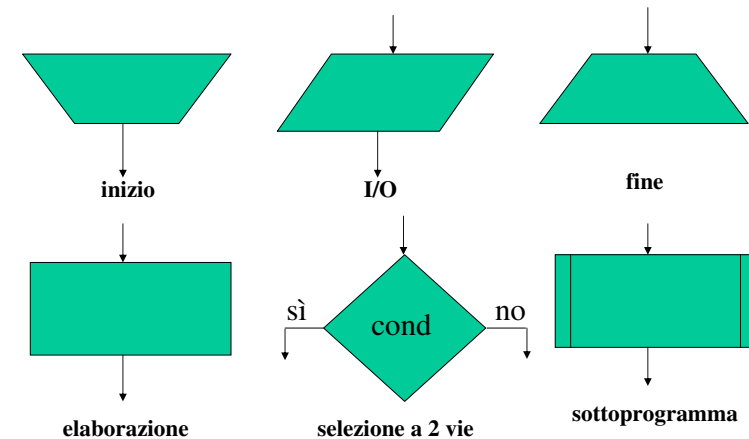
Problema: Determinare il maggiore di due numeri interi

1. Leggi un valore dall'esterno e inseriscilo nella variabile x
2. Leggi un secondo valore e inseriscilo nella variabile y
3. Calcola $x-y$ e assegna il risultato a d
4. Se d è diverso da 0 allora vai al passo 5 altrimenti vai al passo 8
5. Se d è maggiore di 0 allora vai al passo 6 altrimenti vai al passo 7
6. Stampa/visualizza la frase "Il massimo è ..." seguita dal valore contenuto in x e vai al passo 9
7. Stampa/visualizza la frase "Il massimo è ..." seguita dal valore contenuto in y e vai al passo 9
8. Stampa/visualizza la frase "I due numeri sono uguali!"
9. Termina l'esecuzione

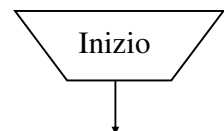
I diagrammi di flusso

- Termini equivalenti: *diagrammi a blocchi*, *schemi a blocchi*, *flowchart*
- Formalismo **grafico** per descrivere algoritmi
- Elementi del linguaggio: simboli per indicare *inizio* e *fine*, operazioni di *ingresso/uscita* (I/O), *elaborazioni*, *condizioni*, *sottoprogrammi*

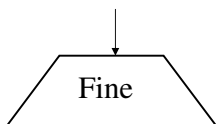
La simbologia comunemente utilizzata



Blocchi Inizio e Fine

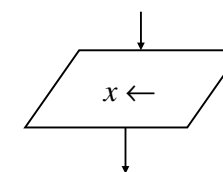


Il blocco '**Inizio**' denota il punto da cui l'esecuzione dell'algoritmo deve iniziare

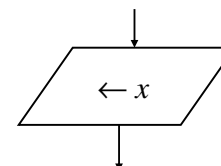


Il blocco '**Fine**' denota il punto in cui l'esecuzione dell'algoritmo termina

Blocco di I/O



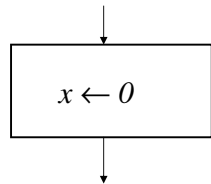
Contiene un'**istruzione di lettura** di un valore dall'esterno (esempio lettura da tastiera) oppure un'**istruzione di scrittura** verso l'esterno (esempio scrittura a video)



$x \leftarrow$ lettura di un valore e sua memorizzazione nella variabile x

$\leftarrow x$ scrittura (invio) del valore di x all'esterno

Blocco di Elaborazione



Contiene un'*istruzione di assegnamento* come ad esempio:

$x \leftarrow y$

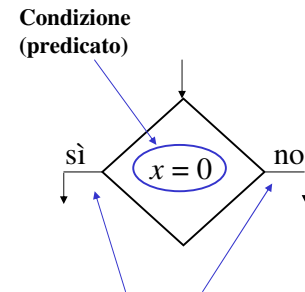
$x \leftarrow x+1$

$x \leftarrow y+z*w$

...

Nei linguaggi di programmazione più comuni il simbolo '=' oppure ':=' viene usato al posto di '←'

Selezione a 2 vie (Blocco decisionale)



Rappresenta un'*istruzione di controllo*

Esempi di *condizioni*:

$x = y$

$x < z$

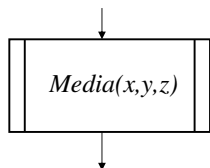
$x \geq z*w$

...

Se la condizione risulta vera **allora** l'esecuzione prosegue lungo l'arco denotato con 'sì' **altrimenti** prosegue lungo l'arco denotato con 'no'

Possono anche indicarsi con V (vero) ed F (falso)

Blocco Sottoprogramma



Contiene un'*istruzione di chiamata di un sottoprogramma* come ad esempio:

$Max(x,y)$




$MCD(y,z)$

...

Nei linguaggi di programmazione più comuni si distingue fra chiamate di procedura e chiamate di funzione

In ogni caso sono sottoprogrammi richiamabili dal programma principale o da altri sottoprogrammi

Collegamenti fra i blocchi

- I blocchi sono collegati fra loro da *archi orientati* 
- L'arco identifica la *sequenza* delle operazioni da compiere 
- La freccia identifica il *flusso di esecuzione* 

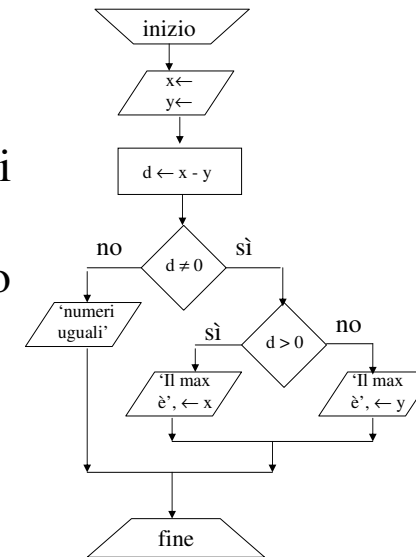
Esempio: Problema 1

Determinare il maggiore di due numeri interi

Algoritmo in linguaggio naturale:

1. Leggi un valore dall'esterno e inseriscilo nella variabile x
2. Leggi un secondo valore e inseriscilo nella variabile y
3. Calcola $x-y$ e assegna il risultato a d
4. Se d è diverso da 0 allora vai al passo 5 altrimenti vai al passo 8
5. Se d è maggiore di 0 allora vai il passo 6 altrimenti esegui il passo 7
6. Stampa/visualizza la frase "Il massimo è ..." seguita dal valore contenuto in x e vai al passo 9
7. Stampa/visualizza la frase "Il massimo è ..." seguita dal valore contenuto in y e vai al passo 9
8. Stampa/visualizza la frase "I due numeri sono uguali!"
9. Termina l'esecuzione

Diagramma di flusso dell'algoritmo che risolve il problema 1



Esecuzione passo passo del programma

- 1 Lettura di un numero e memorizzazione nella variabile x (supponiamo che il numero acquisito sia 8, quindi in memoria si avrà $x=8$)
Letture di un numero e memorizzazione nella variabile y (supponiamo che il numero acquisito sia 6, quindi in memoria si avrà $y=6$)
- 2 Calcolo di $x-y$ e risultato in $d \rightarrow d = 2$
- 3 Controllo se $d \neq 0 \rightarrow$ è vero
- 4 Controllo se $d > 0 \rightarrow$ è vero
- 5 Stampo/visualizzo "Il massimo è 8"
- 6 Fine

Esempio: Problema 2

Calcolo del Massimo Comune Divisore (MCD) di due interi x ed y

Individuazione della procedura di risoluzione:

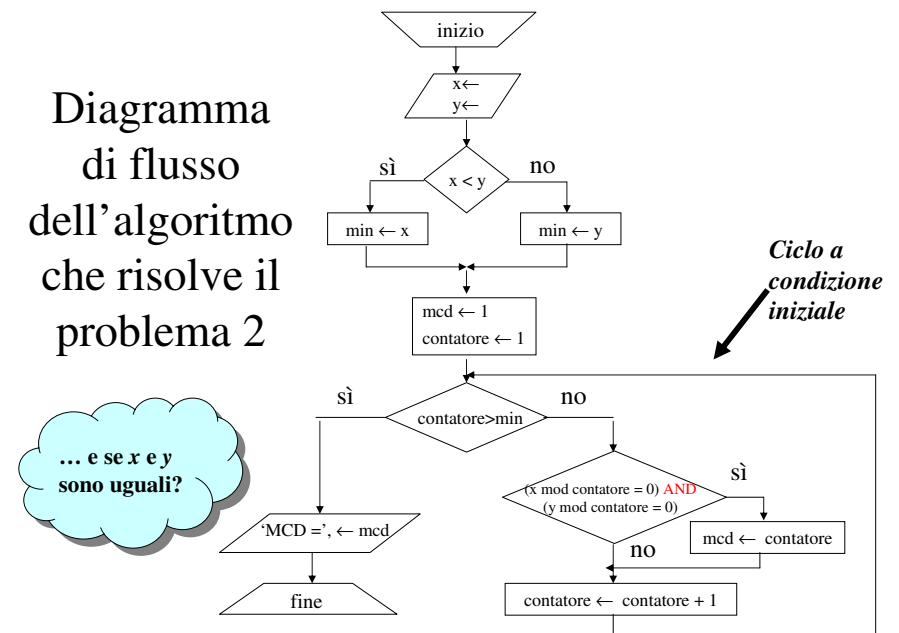
Fai la scansione di tutti i numeri compresi fra 1 e il minimo tra x ed y . Per ognuno stabilisci se è un divisore comune ad x ed y (come? b è divisore di a se $a \bmod b = 0$, ovvero se la divisione di a per b dà come resto 0). Ogni volta che un numero preso in considerazione risulta divisore sia di x che di y memorizzalo come attuale MCD in una variabile all'inizio posta a 1. Alla fine della scansione la variabile conterrà il valore desiderato.

Algoritmo per il Massimo Comune Divisore

Descrizione dell'algoritmo in linguaggio naturale:

1. Leggi un valore dall'esterno e inseriscilo nella variabile x
2. Leggi un valore dall'esterno e inseriscilo nella variabile y
3. Se $x < y$ allora $min \leftarrow x$ altrimenti $min \leftarrow y$
4. $mcd \leftarrow 1$
5. $contatore \leftarrow 1$
6. Se $contatore > min$ allora vai al passo 11 altrimenti vai al passo 7
7. Se $((x \bmod contatore = 0) \text{ e } (y \bmod contatore = 0))$ allora vai al passo 8 altrimenti vai al passo 9
8. $mcd \leftarrow contatore$
9. $contatore \leftarrow contatore + 1$
10. Torna al passo 6
11. Stampa "MCD =" seguito dal valore in mcd
12. Fine

Diagramma di flusso dell'algoritmo che risolve il problema 2



Esecuzione passo passo dell'algoritmo

1. Lettura di due numeri e memorizzazione nelle variabili x e y (supponiamo di acquisire 6 e 9, quindi in memoria si avrà $x=6$ e $y=9$)
2. Controllo se $x < y \rightarrow$ è vero
3. $min \leftarrow 6$
4. $mcd \leftarrow 1$, $contatore \leftarrow 1$
5. controllo se $(contatore > min) \rightarrow$ non è vero
6. controllo se $((x \bmod contatore = 0) \text{ e } (y \bmod contatore = 0))$, cioè se 6 è divisibile per 1 e 9 è divisibile per 1 \rightarrow è vero
7. $mcd \leftarrow 1$
8. $contatore \leftarrow 2$
9. controllo se $(contatore > min) \rightarrow$ non è vero
10. controllo se $((x \bmod contatore = 0) \text{ e } (y \bmod contatore = 0))$, cioè se 6 è divisibile per 2 e 9 è divisibile per 2 \rightarrow non è vero

Esecuzione passo passo dell'algoritmo (cont.)

11. $contatore \leftarrow 3$
12. controllo se $(contatore > min) \rightarrow$ non è vero
13. controllo se $((x \bmod contatore = 0) \text{ e } (y \bmod contatore = 0))$, cioè se 6 è divisibile per 3 e 9 è divisibile per 3 \rightarrow è vero
14. $mcd \leftarrow 3$
15. $contatore \leftarrow 4$
16. controllo se $(contatore > min) \rightarrow$ non è vero
17. controllo se $((x \bmod contatore = 0) \text{ e } (y \bmod contatore = 0))$, cioè se 6 è divisibile per 4 e 9 è divisibile per 4 \rightarrow non è vero
18. $contatore \leftarrow 5$
19. controllo se $(contatore > min) \rightarrow$ non è vero
20. controllo se $((x \bmod contatore = 0) \text{ e } (y \bmod contatore = 0))$, cioè se 6 è divisibile per 5 e 9 è divisibile per 5 \rightarrow non è vero

Esecuzione passo passo dell'algoritmo (cont.)

- 21 $\text{contatore} \leftarrow 6$
- 22 controllo se $(\text{contatore} > \text{min}) \rightarrow$ non è vero
- 23 controllo se $((x \bmod \text{contatore} = 0) \text{ e } (y \bmod \text{contatore} = 0))$, cioè se
6 è divisibile per 6 e 9 è divisibile per 6 \rightarrow non è vero
- 24 $\text{contatore} \leftarrow 7$
- 25 controllo se $(\text{contatore} > \text{min}) \rightarrow$ è vero
- 26 Stampa "MCD = 3" (cioè stampo il valore nella variabile *mcd*)

Esercizio:

Rifare l'esecuzione passo
passo assumendo di
acquisire $x = 18$ e $y = 12$