



Laboratorio di Programmazione

Laurea in Ingegneria Civile e Ambientale

Algoritmi e Programmazione

Stefano Cagnoni

Il problema di fondo

■ Descrizione di un problema

☑ Individuazione di una soluzione

- Quale è il giusto punto di partenza? Cioè, di quali dati abbiamo bisogno ?
- Quali metodologie o tecniche utilizzare?
- In quale ordine eseguire le operazioni consentite da tali tecniche ?

2

Algoritmo

- Dall'arabo al-Khuwarizmi, e dal greco arithmós
- Un algoritmo è un metodo generale che risolve in un **tempo finito** e con una **sequenza finita** di passi **qualsiasi istanza** di un dato problema.

3

Algoritmo: esempio

Problema generale: Calcolare la somma di un numero arbitrario N di numeri

Se ho a disposizione l'operazione di somma fra 2 numeri posso risolvere il problema per *qualsiasi* valore di N con una *stessa* sequenza di operazioni: infatti, sommare un nuovo numero alla somma dei precedenti è *ancora una somma fra due numeri!*

Ad ogni valore di N corrisponde una diversa **istanza** del problema generale, ma tutte le istanze possono essere risolte mediante lo stesso **algoritmo**:

Poni totale=0;

Per N volte: somma un nuovo addendo a totale.

4

Algoritmo

- E' possibile definire algoritmi anche per la risoluzione di problemi non informatici

Esempi:

- Descrivere come raggiungere una destinazione
 - Istruzioni per il montaggio di un mobile
 - Istruzioni per la realizzazione di una torta
- Non sempre la soluzione di un problema può essere descritta tramite un algoritmo.
 - Il calcolo di π non è un algoritmo (non ha fine)
 - Il calcolo di π fino alla decima cifra decimale è un algoritmo

5

Codifica di un algoritmo

- Fase di descrizione (*scrittura*) di un algoritmo attraverso un insieme ordinato (*sequenziale*) di codici (*istruzioni*, ciascuno dei quali rappresenta un'*operazione*), appartenenti a un qualche *linguaggio di programmazione*, che specificano le azioni da compiere e l'ordine in cui devono essere eseguite
- Il prodotto della codifica è un *programma*

6

Programma

- Testo scritto in accordo alla *sintassi* (insieme di regole sulla formazione delle espressioni in un linguaggio) e alla *semantica* (insieme di regole che consentono l'interpretazione del significato del testo) di un *linguaggio di programmazione*
- Un programma rappresenta l'insieme delle istruzioni che descrivono un *processo computazionale*, espresse in un qualche linguaggio
- Un *processo* trasforma un insieme di dati iniziali nei risultati finali mediante una successione di azioni elementari (*operazioni*)

7

Programma

- Un programma può non descrivere un algoritmo (basta che la sequenza di operazioni non sia finita, cioè che il programma non termini) ...
- ... tuttavia può essere molto utile
Es. gestione di un semaforo: non ha fine poiché ripete indefinitamente la stessa sequenza

8

Esecuzione

- L'esecuzione delle azioni nell'ordine specificato dall'algoritmo consente di ottenere i risultati che risolvono il problema a partire dai dati in ingresso

- algoritmo

*Metodo
risolutivo*

- programma

*Codifica del
metodo in un
linguaggio di
programmazione*

9

Algoritmo

- Un algoritmo deve avere le seguenti **proprietà**:
 - **Finitezza**: composto da un numero finito di passi elementari.
 - **Non ambiguità** (determinismo): i risultati non variano in funzione della macchina/persona che esegue l'algoritmo.
 - **Realizzabilità**: deve essere eseguibile con le risorse a disposizione.
 - **Efficienza** (auspicabile): deve eseguire il minimo numero possibile di operazioni

10

Algoritmo

- Per definire un algoritmo è necessario:
 - Condurre un'attenta analisi del problema ed eventualmente suddividere il problema in sottoproblemi più piccoli.
 - Individuare le informazioni disponibili in ingresso (input) e precisare le informazioni che devono essere prodotte dall'elaborazione (output), cioè definire i *dati* di input e di output.
 - Definire completamente e dettagliatamente la sequenza dei passi che portano alla soluzione.

11

Il crivello di Eratostene

- Si vogliono trovare tutti i numeri primi compresi fra 2 e n (in modo efficiente).
 1. Si costruisce una sequenza ordinata dei numeri fra 2 e n .
 2. Si estrae il primo numero dalla sequenza. E' necessariamente un numero primo.
 3. Si eliminano dalla sequenza tutti i multipli del numero estratto al passo 2.
 4. Se la sequenza non è vuota si torna al passo 2, altrimenti si termina.

12

Il crivello di Eratostene

Esempio : trovare i numeri primi compresi fra 2 a 20

Sequenza iniziale:

2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20

□ 2 è primo; lo elimino con tutti i suoi multipli:

3,5,7,9,11,13,15,17,19

□ 3 è primo; lo elimino con tutti i suoi multipli:

5,7,11,13,17,19

□ 5 è primo; lo elimino con tutti i suoi multipli:

...7 ... 11 ... 13 ... 17 ... (li estraggo allo stesso modo)

□ 19 è primo, la sequenza è vuota, **termino**.

13

Diagrammi di flusso (Flow Chart)

- I diagrammi di flusso sono un formalismo grafico per descrivere gli algoritmi.
 - I diagrammi di flusso visualizzano graficamente i passi da cui sono formati gli algoritmi e l'ordine in cui devono essere eseguiti.
 - Un diagramma di flusso è una descrizione *formale*, (cioè rispetta una precisa sintassi), più efficace e meno ambigua di una descrizione a parole.
-

14

Diagrammi di flusso

- Operazioni rappresentabili in un diagramma di flusso:
 - Ingresso/Uscita dati (rappresentate come **schede**)
 - Operazioni sui dati (rappresentate come **rettangoli**)
 - Trasferimento di informazione (assegnazioni)
 - Calcolo di espressioni aritmetiche e logiche
 - Verifica di condizioni (rappresentate come **rombi**)
 - Assunzione di decisioni o cicli (**combinazioni** di rettangoli e rombi)
 - Se... allora ...
 - Ripeti per N volte
 - Ripeti finché
 - Possono utilizzare costanti e variabili
-

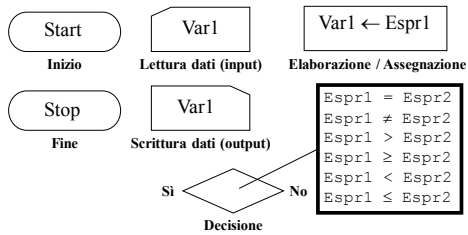
15

Diagrammi di flusso

- Un diagramma di flusso è costituito da due tipi di entità:
 - Nodi
 - rappresentano le operazioni e gli stati di inizio e fine dell'algoritmo
 - Archi orientati (di solito in realtà sono segmenti di retta orientati secondo la direzione di una freccia)
 - Collegano fra loro i nodi, rappresentando con frecce il 'flusso' dei dati, cioè la sequenza delle operazioni: l'istruzione contenuta in un nodo è seguita dall'esecuzione dell'operazione contenuta nel nodo a cui punta l'arco uscente da esso
 - Una struttura di questo tipo è detta *grafo (orientato)*
-

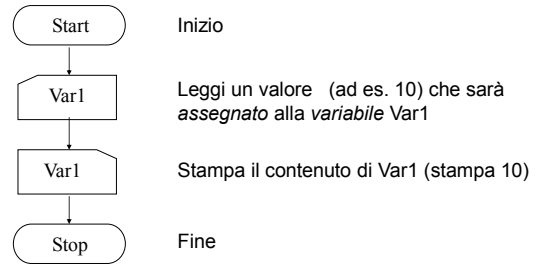
16

Tipi di Nodi



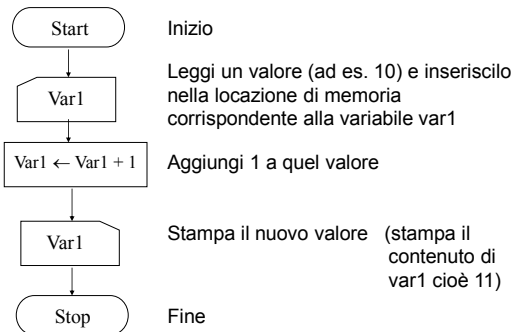
17

Esempio



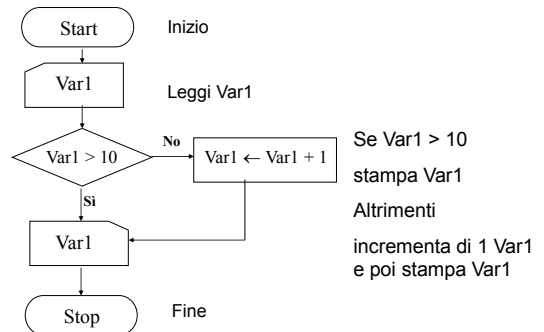
18

Esempio 2



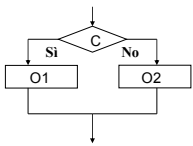
19

Esempio 3



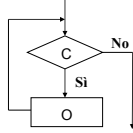
20

Strutture di Controllo



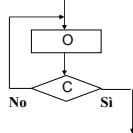
If - Then - Else

Se C è vera esegue l'operazione (o sequenza) O1, altrimenti esegue O2



Ciclo While
Ciclo For

Ripete una stessa operazione (o sequenza) O finché la condizione C resta vera



Ciclo Repeat - Until

Ripete una stessa operazione (o sequenza) O finché la condizione C non diventa vera

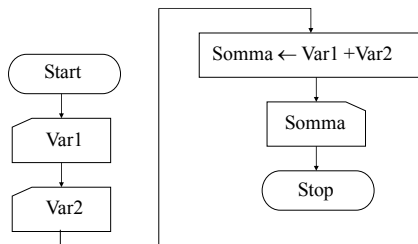
21

Programmazione Strutturata

- Si compone di sequenze di azioni, decisioni (if then, if then else) e cicli (do-while, repeat until).
- Ogni diagramma ha esattamente un ingresso ed una uscita.
- Ogni azione può essere
 - Un'operazione semplice
 - Un'azione composta da altri diagrammi strutturati

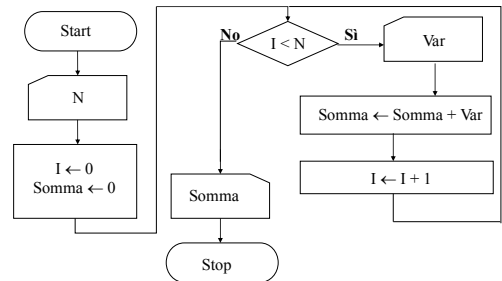
22

Esempio: Somma di due Numeri



23

Esempio: Somma di N Numeri



24