

# STUDIO DI UN ALGORITMO

## INTRODUZIONE

Un algoritmo è una sequenza ordinata di azioni che devono essere eseguite per ottenere il risultato voluto.

Per la risoluzione degli algoritmi bisogna seguire determinati punti:

- 1-Individuare la matrice  $\mathbf{A}$  e il vettore dei termini noti  $\mathbf{v}$ ;
- 2-Scrivere il sistema nella forma  $\mathbf{Ax} = \mathbf{v}$ ;
- 3-Inserire la matrice  $\mathbf{A}$  nel computer (in MATLAB);
- 4-Inserire il vettore  $\mathbf{v}$  nel computer (in MATLAB);
- 5-Ricavare la formula con la quale ricavo il vettore  $\mathbf{x} : \mathbf{x} = \mathbf{A}^{-1} * \mathbf{V}$ ;
- 6-Inserire o ricavare  $\mathbf{A}^{-1}$  con il computer  $\mathbf{B} = \text{inv } \mathbf{A}$ ;
- 7-Scrivere  $\mathbf{x} = \mathbf{B} * \mathbf{v}$  con MATLAB;
- 8-verificare che  $\mathbf{x}$  sia corretto: il risultato deve essere uguale al vettore dei termini noti  $\mathbf{v}$ .

Per una corretta progettazione di algoritmi è utile seguire una certa sequenza ordinata di azioni:

- Stabilire (prima di effettuare il calcolo) chiaramente le soluzioni che dovrà avere l'algoritmo;
- Analizzare i dati posseduti;
- Progettare l'algoritmo utilizzando il metodo TOP-DOWN o BOTTOM-UP;
- Formalizzare l'algoritmo attraverso il FLOW-CHART (vedi "RAPPRESENTAZIONE DI UN SISTEMA");
- Tradurre l'algoritmo progettato con un linguaggio di progettazione adeguato, in questo caso MATLAB.

Il metodo **TOP-DOWN** è un tipo di studio di un problema che inizia con l'analisi del problema che tende ad individuare come prima cosa le caratteristiche fondamentali dell'applicazione, suddividendole quindi in strutture più semplici, per poi passare ad esaminare queste, suddividendole a loro volta in altre strutture, riducendo così l'analisi dello studio dell'applicazione allo studio dei semplici problemi.

Il metodo **BOTTOM-UP** consiste nello studio iniziale delle singole parti elementari costituenti l'applicazione. Unendo poi tutti gli elementi si studia l'applicazione complessiva.

Il metodo dei diagrammi di flusso **FLOW-CHART** sono i linguaggi utilizzati per gli algoritmi; essi sono adatti alla rappresentazione grafica di sistemi sequenziali (un processo è sequenziale quando l'esecuzione di una qualsiasi azione non può mai sovrapporsi all'esecuzione di un'altra), e sono convertibili in più programmi codificati con diversi linguaggi di programmazione.

L'unico difetto di questi diagrammi di flusso è la dimensione: sono molto grandi e spesso non entrano in un foglio.

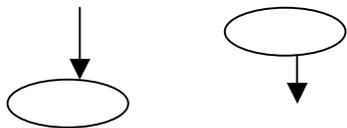
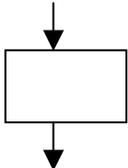
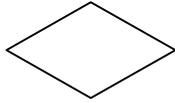
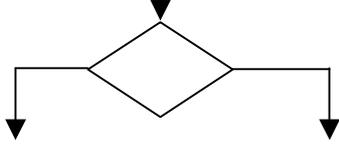
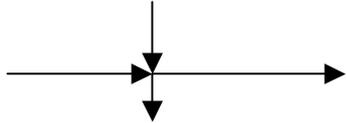
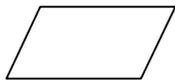
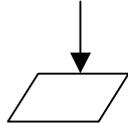
## Osservazioni

Il linguaggio degli algoritmi non e' un linguaggio di programmazione, può essere rappresentato con due metodi:

**GRAFICO:** sviluppando diagrammi di flusso, con carte strutturate, con rappresentazioni ad albero.

**LINEARE:** descrivendolo a parole.

## Simboli che si utilizzano per rappresentare il diagramma di flusso

BLOCCHI ELEMENTARI	SIGNIFICATO	INTERCONNESSIONE
	LIMITE DI UN PROCESSO O OPERAZIONE	
	BLOCCO DI AZIONE	
	SIMBOLO DECISIONALE	
	INTERCOMUNICAZIONE	
	COMUNICAZIONE CON L'ESTERNO	

Gli elementi di base per la realizzazione di un algoritmo sono:

- Blocco iniziale;
- Blocco finale;
- Numero finito di blocchi di azione;
- Numero finito di blocchi di controllo.

## **Il teorema di JACOPINI**

Il teorema di Jacopini enuncia che: *“Dove un processo  $P$  ed un grafico che lo descrive è sempre possibile determinare è sempre possibile determinare un processo  $Q$  che sia descrivibile tramite un grafico di flusso strutturale.”*