UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Fondamenti di Informatica

Ing. Alba Amato, PhD

alba.amato@unina2.it

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

ALGORITMI E PROGRAMMI

Lucidi tratti da:

Alla scoperta dei fondamenti dell'informatica. Un viaggio nel mondo dei bit
di Angelo Chianese, Vincenzo Moscato, Antonio Picariello

capitolo 3 -par. 3.1, 3.2, 3.3, 3.5

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Informatica e Trattamento delle Informazioni

Informatica = studio sistematico dei *processi* che servono al trattamento delle informazioni o più in generale della definizione della soluzione di problemi assegnati.

- –analisi dettagliata di ciò che serve al trattamento dell'informazione,
- –progetto di una soluzione applicabile alla generazione di informazioni prodotte da altre informazioni,
- –verifica della correttezza e della efficienza della soluzione pensata,
- –manutenzione della soluzione nella fase di funzionamento in esercizio

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Informatica e Studio degli Algoritmi

"algoritmo"

- introdotto nella matematica per specificare la sequenza precisa di operazioni il cui svolgimento è necessario per la soluzione di un problema assegnato.
- si basa sulla possibilità che le direttive utilizzate per risolvere un fissato problema siano possibili ovvero eseguibili materialmente→ necessità di un esecutore

Informatica e studio sistematico degli algoritmi.

- Il calcolatore è tra tutti gli esecutori di algoritmi (compreso l'uomo) quello che si mostra più potente degli altri e con una potenza tale da permettere di gestire quantità di informazioni altrimenti non trattabili.

Lo studio dell'Informatica considera quindi il computer come uno scienziato utilizza il proprio microscopio: uno strumento per provare le proprie teorie e, nel caso specifico, verificare i propri ragionamenti o algoritmi

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

La soluzione dei Problemi

- La descrizione del problema non fornisce, in generale, indicazioni sul metodo risolvente; anzi in alcuni casi presenta imprecisioni e ambiguità che possono portare a soluzioni errate.
  - per alcuni problemi non esiste una soluzione;
  - alcuni problemi, invece, hanno più soluzioni possibili; e quindi bisogna studiare quale tra tutte le soluzioni ammissibili risulta quella più vantaggiosa sulla base di un insieme di parametri prefissati (costo della soluzione, tempi di attuazione, risorse necessarie alla sua realizzazione, etc.)
  - per alcuni problemi non esistono soluzioni eseguibili in tempi ragionevoli e quindi utili.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempi

Preparare una torta alla frutta

non si riesce a ricavare alcuna indicazione sulla ricetta da seguire che, tra l'altro, non è facile individuare in un libro di cucina per la sua formulazione generica

Risolvere le equazioni di secondo grado

…un noto e semplice problema di analisi matematica per il quale si conosce chiaramente il procedimento risolvente

Individuare il massimo tra tre numeri

…un esempio di problema impreciso ed ambiguo in quanto non specifica se va cercato il valore massimo o la sua posizione all'interno della terna dei tre numeri assegnati

Calcolare le cifre decimali di π

... un problema con una soluzione nota che però non arriva a terminazione in quanto le cifre da calcolare sono infinite

Inviare un invito ad un insieme di amici

….. si può osservare che esistono sia soluzioni tradizionali basate sulla posta ordinaria, che soluzioni più moderne quali i messaggi SMS dei telefoni cellulari o i messaggi di posta elettronica

Bisogna quindi scegliere la soluzione più conveniente: ad esempio quella che presenta un costo più basso

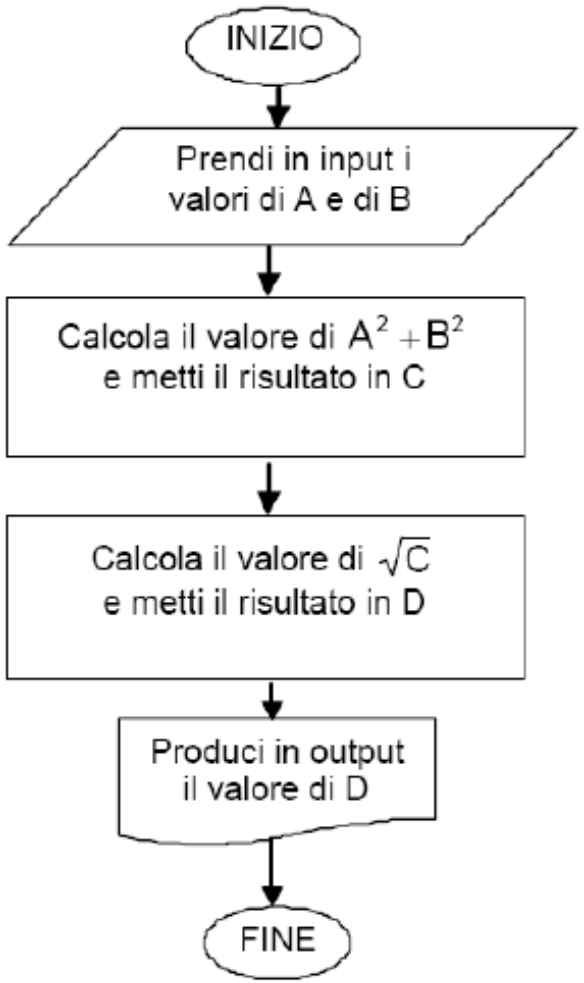
Individuare le tracce del passaggio di extraterrestri

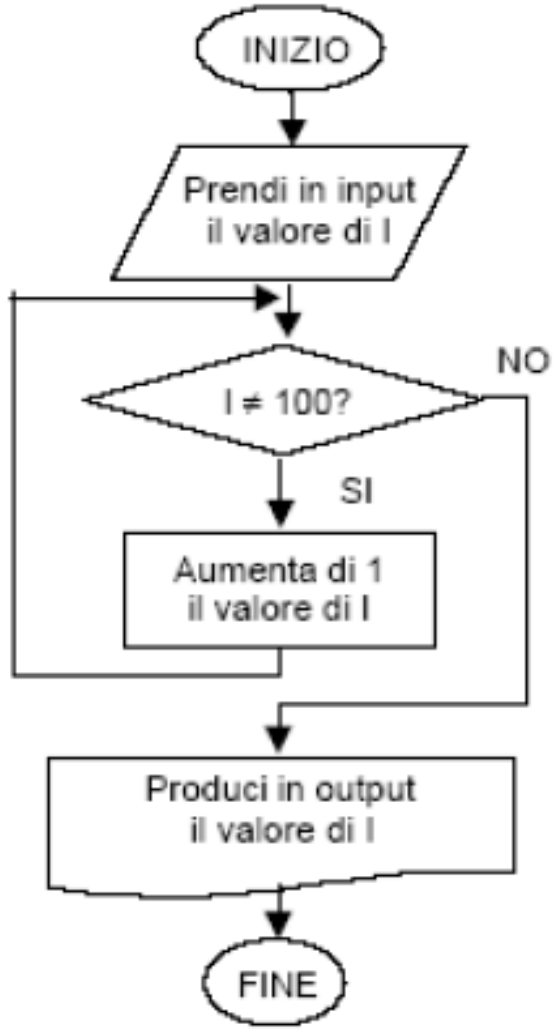
…. chiaro esempio di problema che non ammette soluzione: o, come si dice, non risolvibile

Esercizio 1:

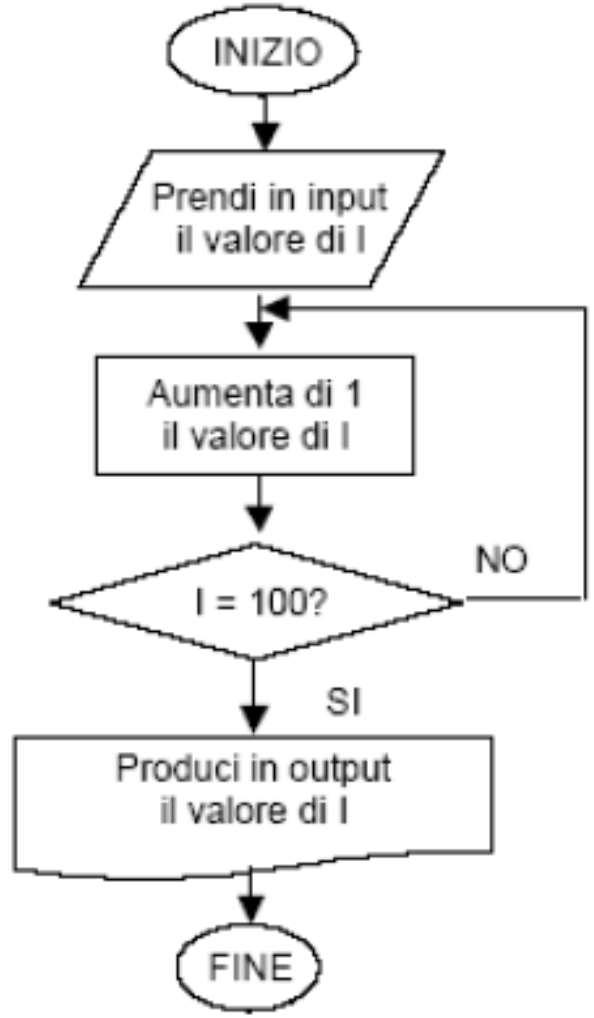
Teorema di Pitagora: Calcolare il valore dell'ipotenusa dati i due cateti:

1. prendi in input la lunghezza A del primo cateto
2. prendi in input la lunghezza B del secondo cateto
3. calcola il quadrato di A
4. calcola il quadrato di B
5. somma i due quadrati
6. estrai la radice quadrata del valore così ottenuto
7. produci in output quest'ultimo valore





a)

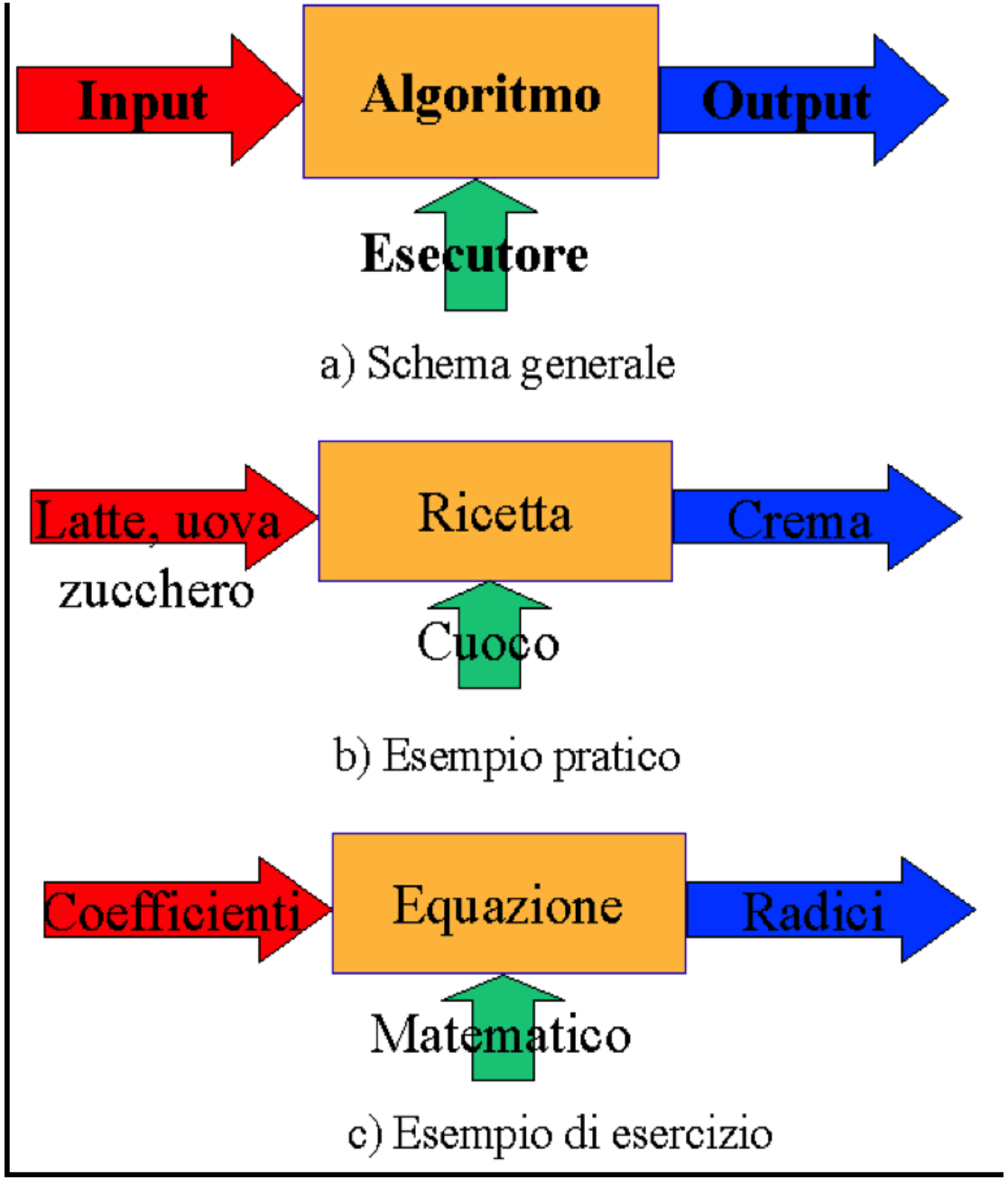


b)

Che problema hanno questi algoritmi? Vanno in loop.

elementi che concorrono alla soluzione dei problemi.

- l'algoritmo, ossia un testo che prescrive un insieme di operazioni od azioni eseguendo le quali è possibile risolvere il problema assegnato.
- l'esecutore, cioè l'uomo o la macchina in grado di risolvere il problema eseguendo l'algoritmo.
- le informazioni di ingresso
- le informazioni di uscita



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Algoritmo ed Esecutore

algoritmo

- …un testo che prescrive un insieme di operazioni od azioni eseguendo le quali è possibile risolvere il problema assegnato.
- Se si indica con **istruzione** la prescrizione di una **singola operazione**, allora l'algoritmo è un insieme di istruzioni da svolgere secondo un ordine prefissato;

esecutore

- …. l'uomo o la macchina in grado di risolvere il problema eseguendo l'algoritmo.
- Se un algoritmo è un insieme di istruzioni da eseguire secondo un ordine prefissato, allora l'esecutore non solo deve comprendere le singole istruzioni ma deve essere anche capace di eseguirle.
- ……una dopo l'altra secondo un ordine rigidamente **sequenziale** che impone l'inizio dell'esecuzione di una nuova istruzione solo al termine di quella precedente, oppure più istruzioni contemporaneamente svolgendole in **parallelo**;

informazioni di ingresso (anche dette input)

- …le informazioni che devono essere fornite affinché avvengano le trasformazioni desiderate;

informazioni di uscita (anche dette output)

- ... i risultati prodotti dall'esecutore del dato algoritmo.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Algoritmi ed Automi

Un algoritmo è fatto di passi in cui avviene una qualche elaborazione, che può essere vista come una funzione:

Y=F(X)

in cui X sono i dati iniziali da elaborare, Y i dati finali o risultati e F è la regola di trasformazione.

Una macchina che esegue algoritmi può essere rappresentata in maniera astratta con un **automa a stati finiti**

- In ogni istante la macchina si trova in una particolare condizione di funzionamento detta **stato**
- A seconda delle elaborazioni compiute in quello stato e a seconda degli eventuali dati in ingresso, la macchina può passare in un nuovo stato

L'automa è uno dei modelli fondamentali dell'informatica

- E' applicabile a qualsiasi sistema che evolve nel tempo per effetto di sollecitazioni esterne.
- Ogni sistema se soggetto a sollecitazioni in ingresso risponde in funzione della sua situazione attuale eventualmente emettendo dei segnali di *uscita*, l'effetto della sollecitazione in ingresso è il mutamento dello stato del sistema stesso.
- Il sistema ha sempre uno stato iniziale di partenza da cui inizia la sua evoluzione.
- Può terminare in uno stato finale dopo aver attraversato una serie di stati intermedi.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Automa a Stati Finiti

- Un automa M (a stati finiti) può essere definito da una quintupla di elementi (Q,I,U,t,w) dove:
  - Q è un insieme finito di **stati** interni caratterizzanti l'evoluzione del sistema;
  - I è un insieme finito di sollecitazioni in **ingresso**;
  - U è un insieme finito di **uscite**;
  - t è la **funzione di transizione** che trasforma il prodotto cartesiano Q×I in Q (t: QxI → Q)
  - w è la **funzione di uscita** che trasforma Q×I in U (w: QxI → U).

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

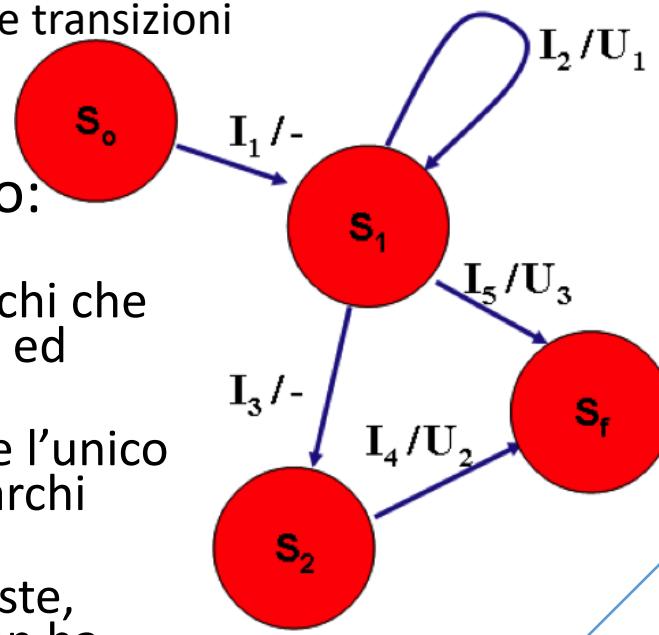
Rappresentazione a Grafo

Grafo

- un cerchio per rappresentare gli stati del sistema
- archi orientati ad indicare le transizioni

Nel grafo si individuano:

- gli stati intermedi rappresentati da cerchi che hanno archi entranti ed uscenti,
- lo stato iniziale come l'unico cerchio che non ha archi entranti;
- lo stato finale, se esiste, come cerchio che non ha archi uscenti.



$Q = (S_o, S_1, S_2, S_f)$
$I = (I_1, I_2, I_3, I_4, I_5)$
$U = (U_1, U_2, U_3)$

I/S	S_o	S_1	S_2	S_f
I_1	S_1 /-			
I_2		S_1 /U_1		
I_3		S_2 /-		
I_4			S_f /U_2	
I_5		S_f /U_3		

Al grafo può essere associata una rappresentazione più matematica in forma tabellare con tante righe quanti sono gli ingressi, e tante colonne quanti sono gli stati.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Macchina per la distribuzione automatica di bibite

Tutti i tipi di bibita costano 2 Euro

La macchina accetta solo monete da 50 centesimi

Si analizzi il comportamento di questa macchina, considerando che ha come ingresso le monete e come uscita la consegna di una bibita

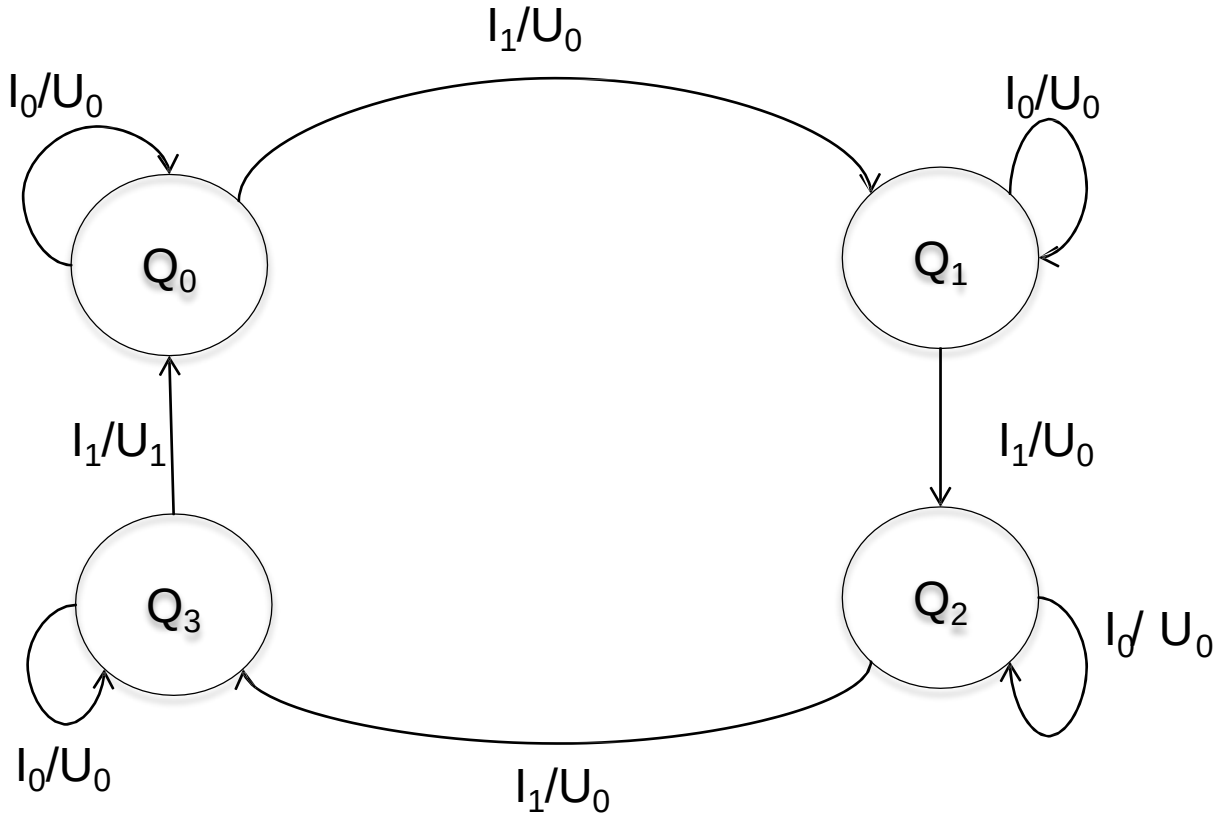
UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Macchina per la distribuzione automatica di bibite

I_0 = nessuna moneta
I_1 = moneta inserita

U_0 = bibita non erogata
U_1 = bibita erogata

Q_0 = 0 centesimi
Q_1 = 50 centesimi
Q_2 = 1 euro
Q_3 = 1 euro e 50 centesimi



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Macchina per la distribuzione automatica di bibite

	Q_0	Q_1	Q_2	Q_3
I_0	Q_0/U_0	Q_1/U_0	Q_2/U_0	Q_3/U_0
I_1	Q_1/U_0	Q_2/U_0	Q_3/U_0	Q_0/U_1

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esercizio 2

Si disegni il grafo di una macchina di stato che, ricevendo in ingresso una sequenza di caratteri dell'insieme
I = [P,A], riconosca la parola "PAPA"

Si riconoscano anche eventuali sequenze sovrapposte (concatenazione)

In uscita si abbia S se si trova la parola

Ad es., se la stringa di ingresso è 'PPPAPAPAAAAA', l'uscita è 'SS'

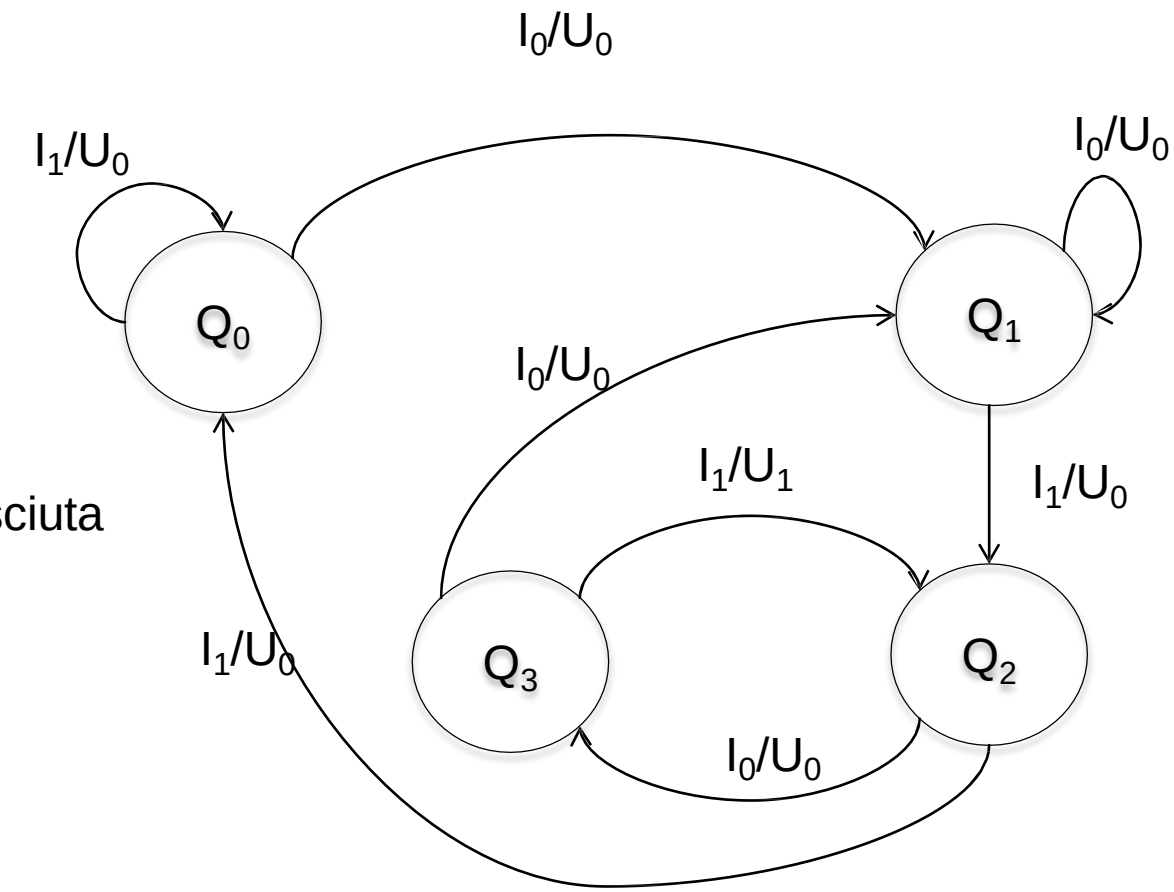
UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esercizio 2

$I_0 = P$
$I_1 = A$

$U_0 = -$
$U_1 = S$

Q_0 = nessuna stringa riconosciuta
$Q_1 = P$
$Q_2 = PA$
$Q_3 = PAP$



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esercizio 2

	Q_0	Q_1	Q_2	Q_3
I_0	Q_1/U_0	Q_1/U_0	Q_3/U_0	Q_1/U_0
I_1	Q_0/U_0	Q_2/U_0	Q_0/U_0	Q_2/U_1

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esercizio 3

Si realizzi un automa a stati finiti che, ricevendo sull'ingresso I una sequenza di bit, segnali con il valore S sull'uscita U (normalmente a N), la presenza di due bit consecutivi a 1

Non considerando le concatenazioni

Per **I = 00111011011110**, deve essere **U = NNNSNNNSNNSNSN**

Considerando anche le concatenazioni

Per **I = 00111011011110**, deve essere **U = NNNSSNNSNNSSSN**

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

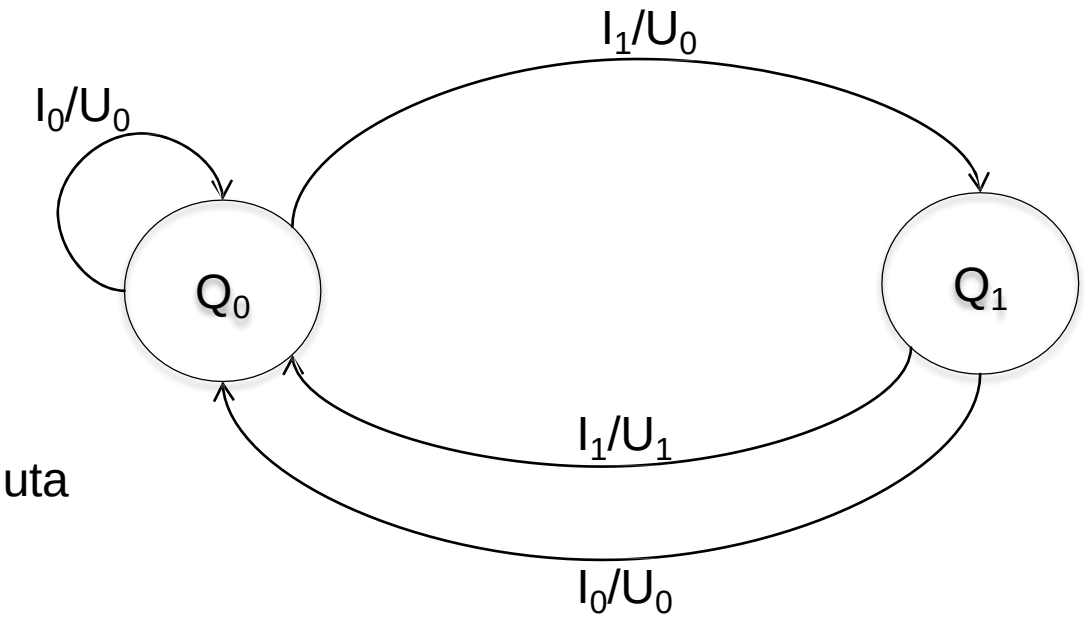
Esercizio 3

Senza concatenazione

$I_0 = 0$
$I_1 = 1$

$U_0 = N$
$U_1 = S$

Q_0 = nessuna stringa riconosciuta
$Q_1 = 1$



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esercizio 3

Senza concatenazione

	Q_0	Q_1
I_0	Q_o/U_0	Q_o/U_0
I_1	Q_1/U_0	Q_0/U_1

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

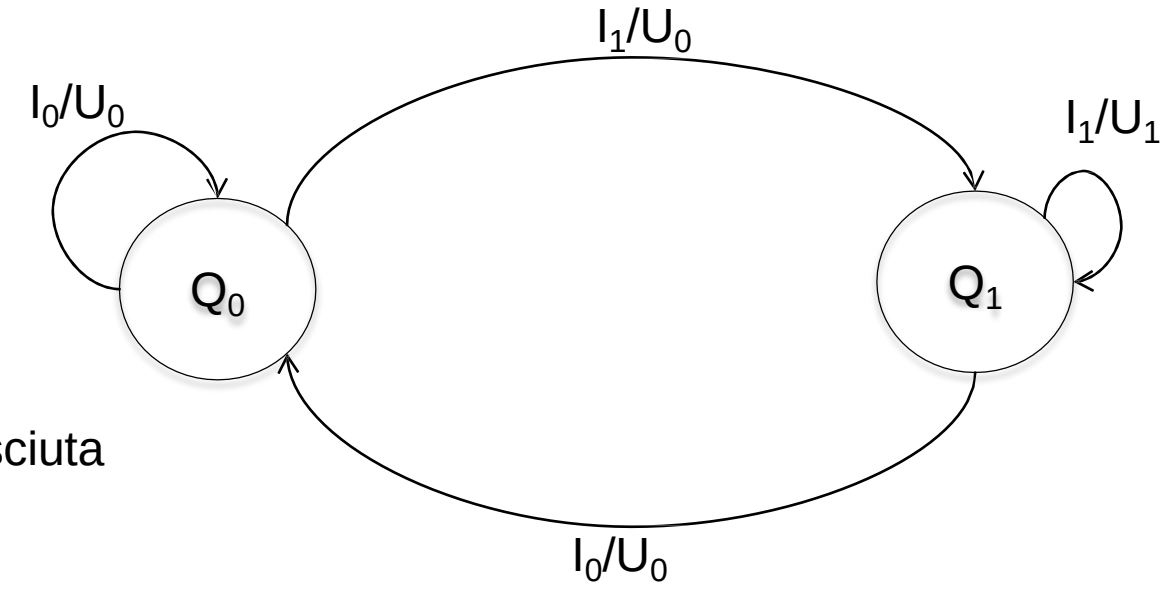
Esercizio 3

Con concatenazione

$I_0 = 0$
$I_1 = 1$

$U_0 = N$
$U_1 = S$

Q_0 = nessuna stringa riconosciuta
$Q_1 = 1$



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esercizio 3

Con concatenazione

	Q_0	Q_1
I_0	Q_o/U_0	Q_o/U_0
I_1	Q_1/U_0	Q_1/U_1

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

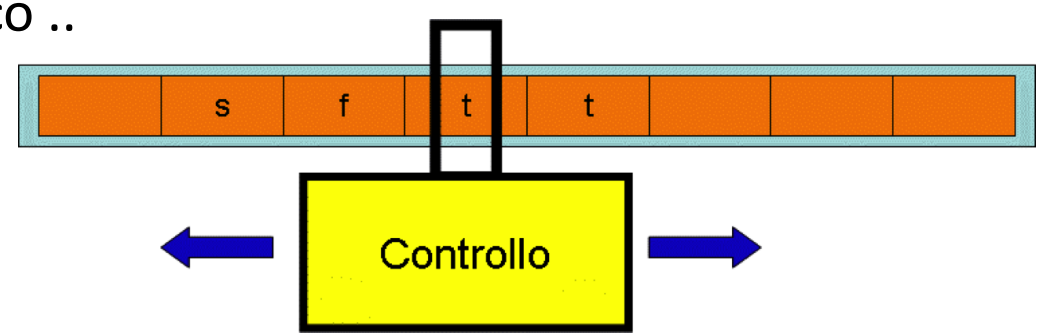
Macchina di Turing

- Il modello di Macchina di Turing è un particolare automa per il quale sono definiti
  - l'insieme degli ingressi e delle uscite come insiemi di simboli
  - è definito un particolare meccanismo di lettura e scrittura delle informazioni.
- È il modello fondamentale della macchina universale, cioè una macchina in grado di realizzare qualsiasi sequenza computabile.
- Il modello permette di **raggiungere risultati teorici sulla calcolabilità e sulla complessità degli algoritmi**.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Macchina di Turing

- E’ un automa con testina di scrittura/lettura su nastro bidirezionale potenzialmente illimitato.
  - Ad ogni istante la macchina si trova in uno stato appartenente ad un insieme finito e legge un simbolo sul nastro.
  - La funzione di transizione, in modo deterministico ..
  - fa scrivere un simbolo
  - fa spostare la testina in una direzione o nell'altra
  - fa cambiare lo stato.

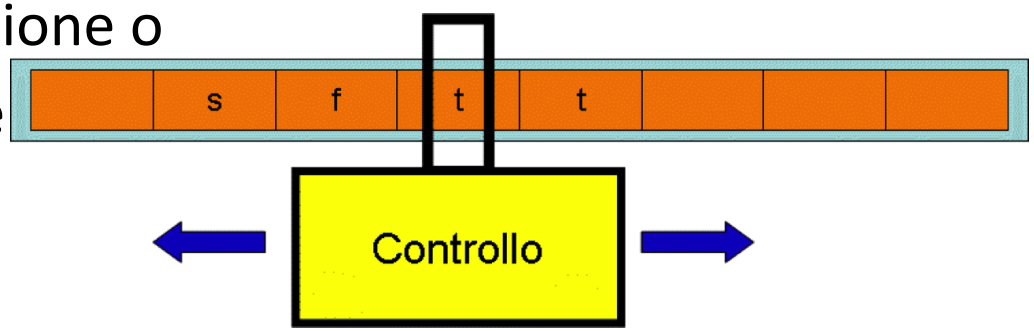


- La macchina si compone di:
  - una memoria costituita da un nastro di dimensione infinita diviso in celle; ogni cella contiene un simbolo oppure è vuota;
  - una testina di lettura scrittura posizionabile sulle celle del nastro;
  - un dispositivo di controllo che, per ogni coppia (stato, simbolo letto) determina il cambiamento di stato ed esegue un’azione elaborativa.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Macchina di Turing: Funzionamento

- l comportamento di una macchina di Turing è molto semplice:
  - inizialmente la macchina si trova in uno stato detto iniziale
  - la stringa di input è scritta sul nastro
  - la testina è posizionata su una cella, per convenzione il carattere più a sinistra della stringa di input
- in base allo stato corrente della macchina di controllo ed al carattere letto o puntato dalla testina si determina lo stato successivo e si compiono le seguenti azioni:
  1. legge, scrive o cancella il carattere dalla cella su cui è posizionata la testina
  2. sposta la testina di una cella a destra o sinistra o lascia la testina ferma
- la macchina continua ad elaborare la stringa sul nastro ed a transire di stato finchè si ferma in uno stato per cui non è definita alcuna azione o
- raggiunge uno stato particolare denominato finale



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Definizione Formale

- è definita dalla quintupla: (**A, S, f_m, f_s, f_d**)
  - **A** è l'insieme finito dei simboli di ingresso e uscita;
  - **S** è l'insieme finito degli stati (di cui uno è quello di terminazione);
  - **F_m** è la funzione di macchina definita come $A \times S \rightarrow A$;
  - **F_s** è la funzione di stato $A \times S \rightarrow S$;
  - **F_d** è la funzione di direzione $A \times S \rightarrow D = \{Sinistra, Destra, Nessuna\}$
- La macchina è capace di:
  - leggere un simbolo dal nastro;
  - scrivere sul nastro il simbolo specificato dalla funzione di macchina;
  - transitare in un nuovo stato interno specificato dalla funzione di stato;
  - spostarsi sul nastro di una posizione nella direzione indicata dalla funzione di direzione.
- La macchina si ferma quando raggiunge lo stato di terminazione

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Macchina di Turing e Algoritmi

- Una macchina di Turing che
  - si arresti
  - trasformi un nastro t in uno t'
- rappresenta l'algoritmo per l'elaborazione Y=F(X), ove Xe Y sono codificati rispettivamente in t e t'.
- Una macchina di Turing la cui parte di controllo è capace di leggere da un nastro anche **la descrizione dell'algoritmo** è **una macchina universale** capace di simulare il lavoro compiuto da un'altra macchina qualsiasi.
  - …leggere dal nastro la descrizione dell'algoritmo richiede di saper interpretare il linguaggio con il quale esso è stato descritto.
- La **Macchina di Turing Universale** è l'**interprete** di un linguaggio e si può definire per qualunque linguaggio

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Tesi di Church e Turing

- **Tesi di Church e Turing**: ***Non esiste alcun formalismo, per modellare una determinata computazione meccanica, che sia più potente della Macchina di Turing e dei formalismi ad essi equivalenti***.
  - Ogni algoritmo può essere codificato in termini di Macchina di Turing ed è quindi ciò che può essere eseguito da una macchina di Turing.
  - Un problema è non risolubile algoritmicamente se nessuna Macchina di Turing è in grado di fornire la soluzione al problema in tempo finito.
  - Se dunque esistono problemi che la macchina di Turing non può risolvere, si conclude che esistono algoritmi che non possono essere **calcolati**.
- Sono problemi **decidibili** quei problemi che possono essere meccanicamente risolvibili da una macchina di Turing;
- sono **indecidibili** tutti gli altri.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Turing vs Von Neumann

- La macchina di Turing e la macchina di von Neumann sono due modelli di calcolo fondamentali per caratterizzare la modalità di descrizione e di esecuzione degli algoritmi.
- La macchina di Von Neumann (1945) fu modellata dalla Macchina di Turing Universale (1936) per ciò che attiene alle sue modalità di computazione.
  - La sua memoria è però **limitata** a differenza del nastro di Turing che ha lunghezza infinita.
  - La macchina di Von Neumann, come modello di riferimento per sistemi non solo teorici, prevede anche la dimensione dell'interazione attraverso i suoi dispositivi di input ed output.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Calcolabilità e Trattabilità

- calcolabilità
  - consente di dimostrare *l'esistenza di un algoritmo* risolvente un problema assegnato ed indipendente da qualsiasi automa
- trattabilità
  - studia la eseguibilità di un algoritmo da parte di un sistema informatico.
- Esistono problemi classificati come risolvibili ma praticamente intrattabili non solo dagli attuali elaboratori ma anche da quelli, sicuramente più potenti, del futuro.
- Sono noti problemi che
  - …. pur presentando un algoritmo di soluzione, non consentono di produrre risultati in tempi ragionevoli neppure se eseguiti dal calcolatore più veloce.
  - ammettono soluzione di per sé lenta e quindi inaccettabile.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Trattabilità e Complessità

- Il concetto di trattabilità è legato a quello di ***complessità computazionale***
  - .. studia i costi intrinseci alla soluzione dei problemi, con l'obiettivo di comprendere le prestazioni massime raggiungibili da un algoritmo applicato a un problema.
- La complessità consente di individuare i problemi risolvibili che siano trattabili da un elaboratore con costi di risoluzione che crescano in modo ragionevole al crescere della dimensione del problema.
- Tali problemi vengono detti trattabili

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Spazio e Tempo

- La complessità di un algoritmo corrisponde a una misura delle risorse di calcolo consumate durante la computazione ed è tanto più elevata quanto maggiori sono le risorse consumate.
- **Complessità spaziale**
  - …. quantità di memoria necessaria alla rappresentazione dei dati necessari all'algoritmo per risolvere il problema;
- **Complessità temporale**
  - …. il tempo richiesto per produrre la soluzione.
- Le misure di complessità possono essere:
  - Statiche: se sono basate sulle caratteristiche strutturali (ad esempio il numero di istruzioni) dell'algoritmo e prescindono dai dati di input su cui esso opera.
  - Dinamiche: se tengono conto sia delle caratteristiche strutturali dell'algoritmo che dei dati di input su cui esso opera

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Complessità e Input

- Un primo fattore che incide sul tempo impiegato dall'algoritmo è la quantità di dati su cui l'algoritmo deve lavorare
  - il tempo di esecuzione è solitamente espresso come una funzione f(n) della dimensione n dei dati di input.
  - Si dirà, ad esempio, che un algoritmo ha un tempo di esecuzione n^2 se il tempo impiegato è pari al quadrato della dimensione dell'input.
- Da sola la dimensione dei dati di input non basta.
- Il tempo di esecuzione dipende anche dalla configurazione dei dati in input oltre che dalla loro dimensione

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Complessità e Input

- analisi del caso migliore
  - …per calcolare il tempo di esecuzione quando la configurazione dei dati presenta difficoltà minime di trattamento.
- analisi del caso peggiore
  - …. per calcolare il tempo di esecuzione quando la configurazione dei dati presenta difficoltà massime di trattamento.
- Si tratta di un'analisi molto utile, perché fornisce delle garanzie sul tempo massimo che l'algoritmo può impiegare
- analisi del caso medio
  - per calcolare il tempo di esecuzione quando la configurazione presenta difficoltà medie di trattamento

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Complessità Asintotica

- Interessa sapere come l'ordine di grandezza del tempo di esecuzione cresce al limite, ossia per dimensioni dell'input sufficientemente grandi quando la funzione *f(n)*tende ai suoi asintoti.
  - Lo studio della condizione asintotica della complessità di un algoritmo permette ulteriormente di trascurare in un algoritmo operazioni non significative concentrando l'attenzione solo su quelle predominanti.
  - dati due algoritmi diversi che risolvono lo stesso problema e presentano due diverse complessità f(n) e g(n), se f(n) è asintoticamente inferiore a g(n) allora esiste una dimensione dell'input oltre la quale l'ordine di grandezza del tempo di esecuzione del primo algoritmo è inferiore all'ordine di grandezza del tempo di esecuzione del secondo.
- La complessità asintotica dipende solo dall'algoritmo, mentre la complessità esatta dipende da tanti fattori legati alla esecuzione dell'algoritmo

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Tipi di Complessità

Si consideri un elaboratore capace di eseguire un milione istruzioni al secondo (1 MIPS)

- A seconda della funzione f(n) asintotica, si possono individuare:
- complessità di tipo polinomiale
  - n(lineare)
  - n^2, n^3, n^5, ….
- Complessità di tipo esponenziale (o, in generale, Non Polinomiale, NP)
  - 2^n, 3^n, ….

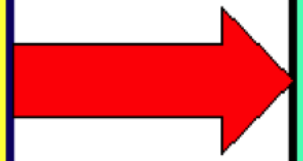
	10	20	30	40	50
n	0,00001 sec	0,00002 sec	0,00003 sec	0,00004 sec	0,00005 sec
n^2	0,0001 sec	0,0004 sec	0,0009 sec	0,0016 sec	0,0025 sec
n^3	0,001 sec	0,008 sec	0,027 sec	0,064 sec	0,125 sec
n^5	0,1 sec	3,2 sec	24,3 sec	1,7 min	5,2 min
2^n	0,001 sec	1,0 sec	17,9 min	12,7 giorni	35,7 anni
3^n	0,059 sec	58 min	6,5 anni	3,855 secoli	200.000.000 secoli

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Descrizione degli Algoritmi

- Dato un problema …
  - …capire preliminarmente se il problema ammette soluzioni
  - …nel caso ne ammetta, individuare un metodo risolutivo (algoritmo)
  - …. esprimere tale metodo in un linguaggio comprensibile all'esecutore a cui è rivolto

La comne osservazione dela evoluzione del mondo degli elaboratori eletronici non deve trarre in ingano. Difati a dispeto del'evoluzione tecnologicha, l'attivita di programmazione di propone come un'arte che è ben vero che muta, ma che non può non contenere i trati antropomorfi di chi genera i programmi e di chi li utilizza.



La comune osservazione della evoluzione del mondo degli elaboratori elettronici non deve trarre in inganno. Difatti a dispetto dell'evoluzione tecnologica, l'attività di programmazione di propone come un'arte che è ben vero che muta, ma che non può non contenere i tratti antropomorfi di chi genera i programmi e di chi li utilizza.

Esempio: scrivere un algoritmo che corregge gli errori lessicali in un testo

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Descrizione degli Algoritmi

SOLUZIONE

```
1     leggi un rigo del testo;
2     per ogni parola del rigo
      fai:
3           se conosci la parola,
            allora controlla come è scritta
            altrimenti fai  una ricerca nel vocabolario
4           se la parola non è riportata in modo corretto
            allora correggila,
5           riscrivi la parola nel testo corretto;
6     ripeti le azioni da 1) a 5)
      fino alla terminazione dei righi del testo.
```

Esempio: scrivere un algoritmo che corregge gli errori lessicali in un testo

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Costrutti

- Si usano naturalmente *costrutti* che fissano l'ordine in cui le diverse azioni devono essere svolte.
  - Il più semplice tra essi è quello che stabilisce che le azioni devono essere svolte una dopo l'altra. (**sequenza**)
  - un altro costrutto stabilisce che alcune azioni devono essere svolte solo se si verificano determinate condizioni (**selezione**)
    - la frase "se la parola è sbagliata, allora correggi, altrimenti non fare niente" prescrive la correzione soltanto in presenza di un errore.
  - un ultimo costrutto dice che alcune azioni devono essere ripetute un numero di volte prestabilito
    - ("per ogni parola del rigo fai")
  - o determinato dal verificarsi di certe condizioni (**iterazione**)
    - ("ripeti le azioni da 1) a 4) fino alla terminazione del testo")
- In tutti gli algoritmi si possono individuare quindi due classi fondamentali di frasi:
  - quelle che prescrivono la esecuzione di determinate operazioni (istruzioni);
  - e quelle che indicano all'esecutore l'ordine in cui tali operazioni devono essere eseguite (strutture di controllo).

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Istruzioni Elementari

- Istruzioni:
  - *elementari* quelle istruzioni che l'esecutore è in grado di comprendere ed eseguire;
  - *non elementari* quelle non note all'esecutore.
- Perché un'istruzione non elementare possa essere eseguita dall'esecutore a cui è rivolta, deve essere specificata in termini più semplici.
- Il procedimento che trasforma una istruzione non elementare in un insieme di istruzioni elementari, prende il nome di *raffinamento o specificazione dell'istruzione non elementare*.
  - Il processo di raffinamento è molto importante. Senza di esso si dovrebbero esprimere gli algoritmi direttamente nel linguaggio di programmazione disponibile

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Caratteristiche delle Istruzioni

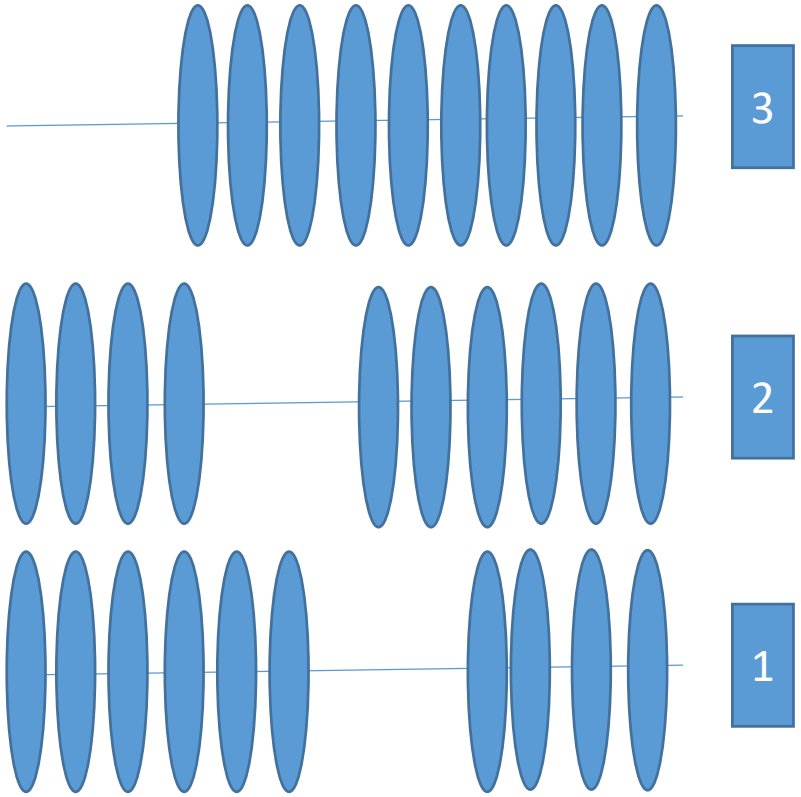
- finitezza:
  - Le operazioni devono avere termine entro un intervallo di tempo finito dall'inizio della loro esecuzione;
- descrivibilità:
  - Le operazioni devono produrre, se eseguite, degli effetti descrivibili, per esempio fotografando lo stato degli oggetti coinvolti sia prima che dopo l'esecuzione dell'operazione;
- riproducibilità:
  - Le operazioni devono produrre lo stesso effetto ogni volta che vengono eseguite nelle stesse condizioni iniziali;
- comprensibilità:
  - Le operazioni devono essere espresse in una forma comprensibile all'esecutore che deve eseguirle.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Un Esempio di Algoritmo: Somma con pallottoliere

Inizializza il pallottoliere

- Sposta tutte le palline a destra
- Inserisci il primo addendo *spostando a sinistra sulla prima riga* un numero di palline pari all'addendo
- Inserisci il secondo addendo *spostando a sinistra sulla seconda riga* un numero di palline pari all'addendo



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Un Esempio di Algoritmo: Somma con pallottoliere

Passo 1: Sposta a destra una pallina sulla prima riga

Passo 2: Sposta a sinistra una pallina sulla terza riga

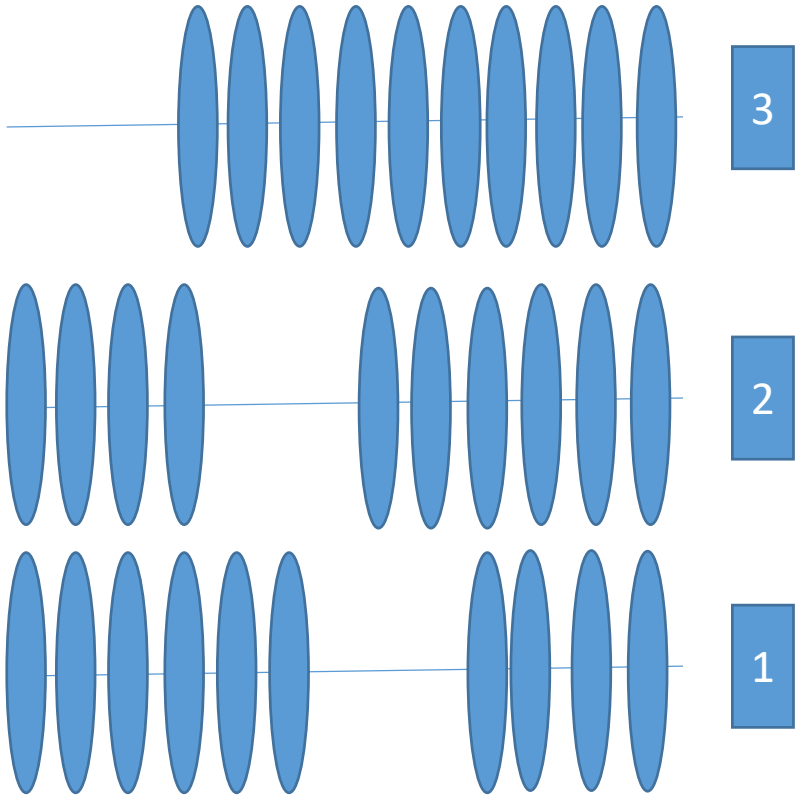
Passo 3: Ritorna a Passo 1 se hai palline a sinistra sulla prima riga

Passo 4: Sposta a destra una pallina sulla seconda riga

Passo 5: Sposta a sinistra una pallina sulla terza riga

Passo 6: Ritorna a Passo 4 se hai palline a sinistra sulla seconda riga

FINE



Risultato finale:

Numero di palline a sinistra sulla terza riga

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

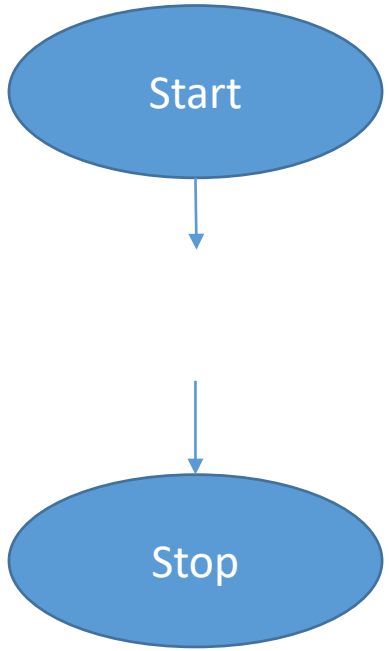
Diagrammi di Flusso

- I ***Diagrammi di flusso*** sono un *linguaggio grafico* per la rappresentazione di un algoritmo
  - Il diagramma è compost da blocchi che rappresentano I passi dell'algoritmo
  - I blocchi hanno forme diverse in base al significato
  - I blocchi contengono testo che dscrive il passo da eseguire
- ***CHI è l'esecutore?***
  - Utili per descrivere un algoritmo più che per eseguirlo…

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso

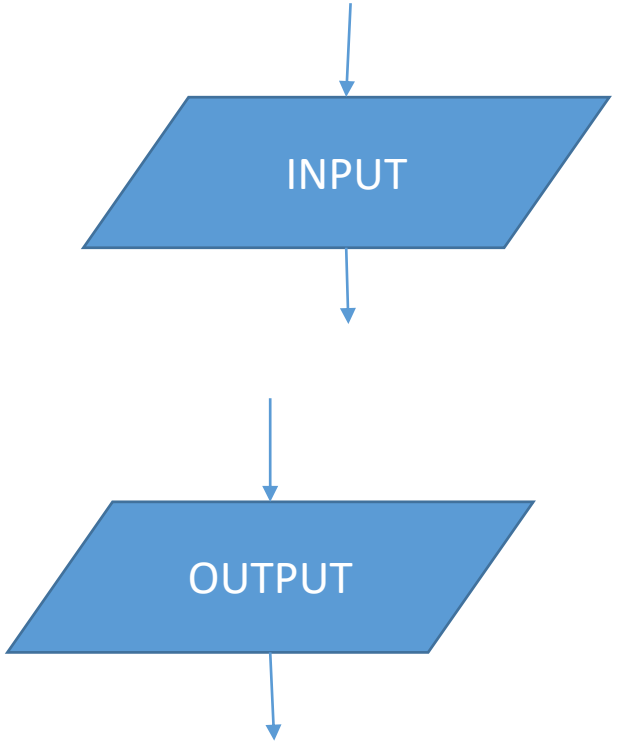
- Blocchi ***Start*** e ***Stop*** vengono utilizzati per indicare da dove il diagramma inizia e quando il diagramma termina.



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso

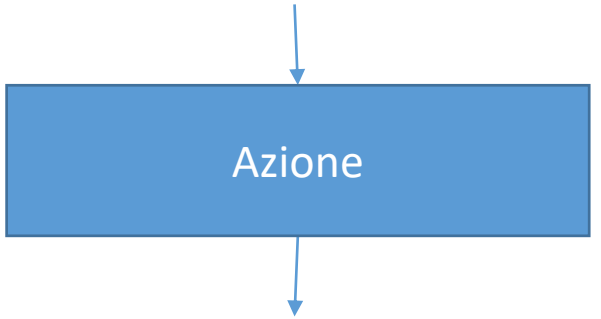
- I Blocchi ***Input*** e ***Output*** vengono utilizzati per indicare la richiesta di dati all'utente



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso

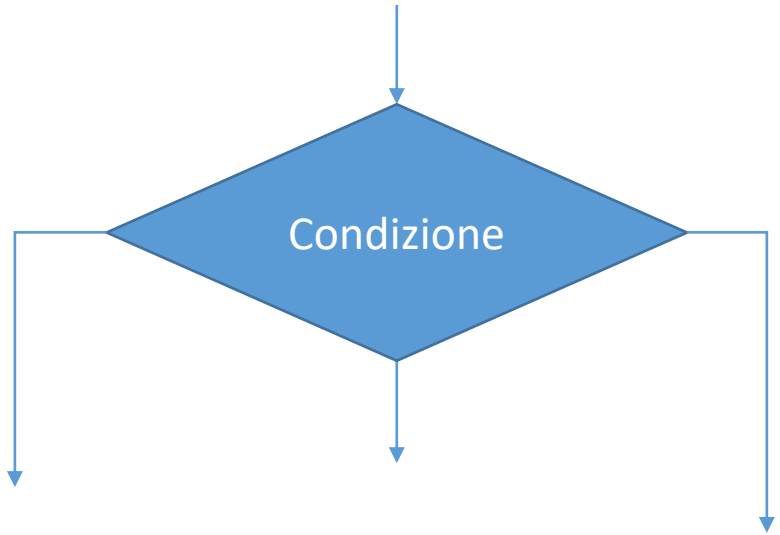
- I Blocchi ***Azione*** vengono utilizzati per indicare una operazione di elaborazione elementare



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso

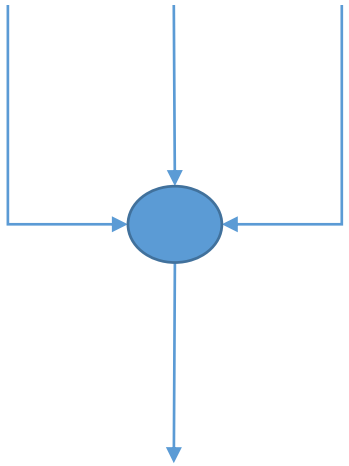
- I Blocchi ***Condizione*** vengono utilizzati per indicare una operazione di scelta
- Ogni ramo di uscita rappresenta una scelta differente



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso

- I Blocchi ***connettore*** vengono utilizzati per indicare che due o più blocchi differenti portano allo stesso blocco



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso

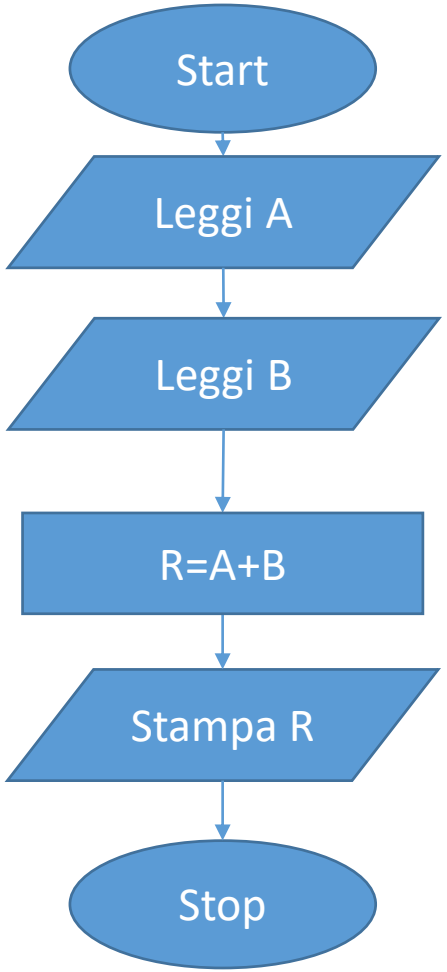
Somma di due numeri

Passo 1: Leggi A

Passo 2: Leggi B

Passo 3: Somma A e B

Passo 4: Stampa il risultato



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso

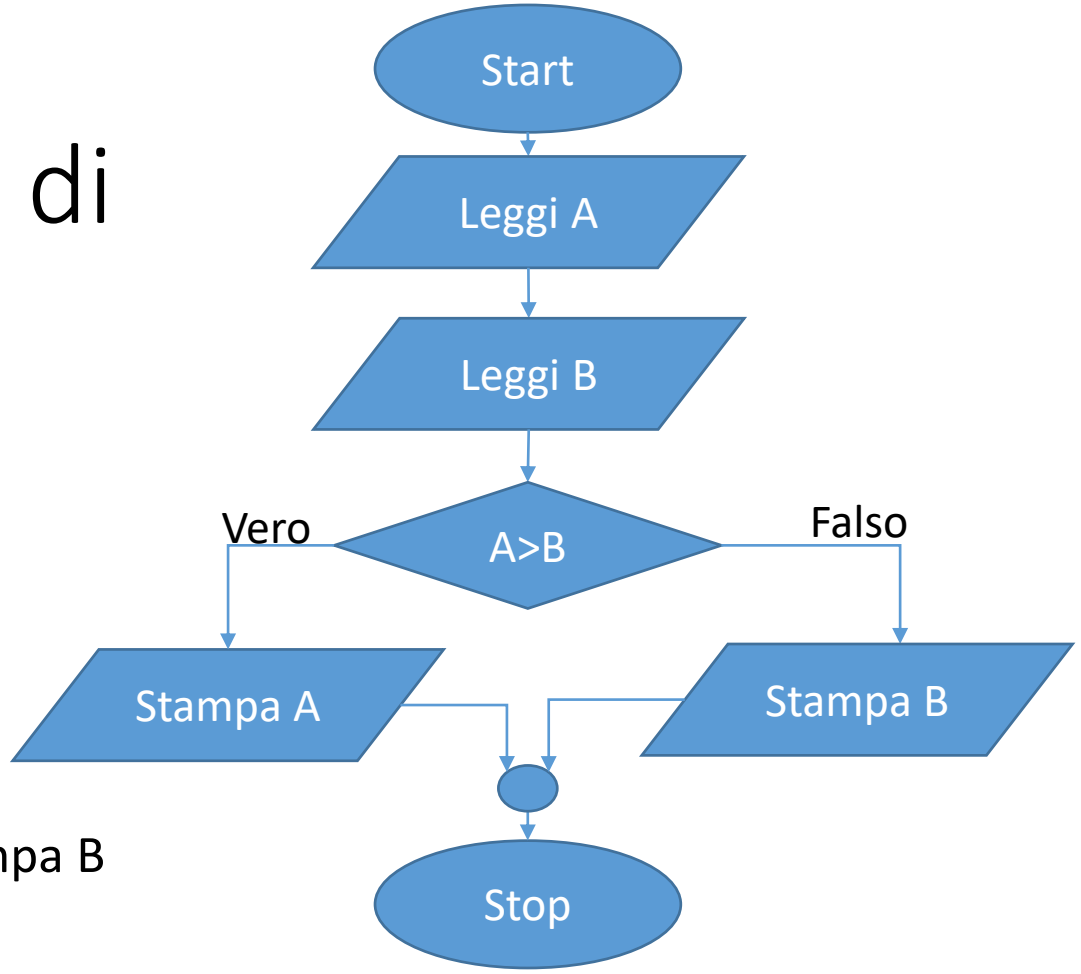
Stampa il maggiore di due numeri

Passo 1: Leggi A

Passo 2: Leggi B

Passo 3: confronta A e B

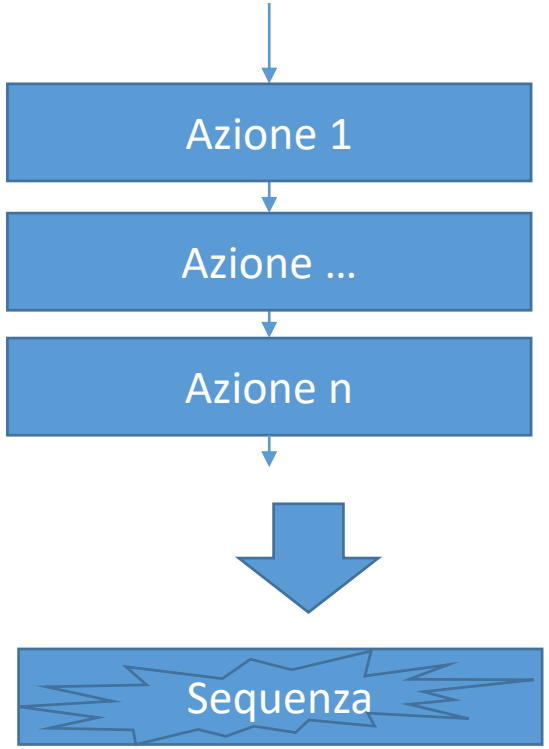
Passo 4: Se A>B stampa A, altrimenti stampa B



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso

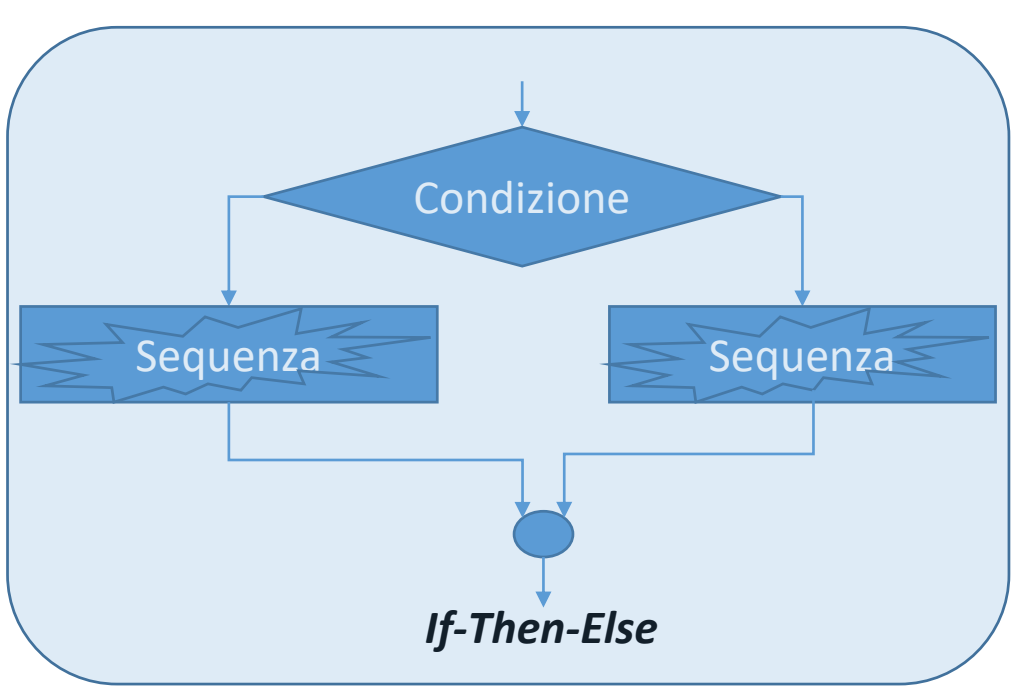
- Il blocco ***sequenza*** viene utilizzato per indicare l'esecuzione consecutiva di blocchi.

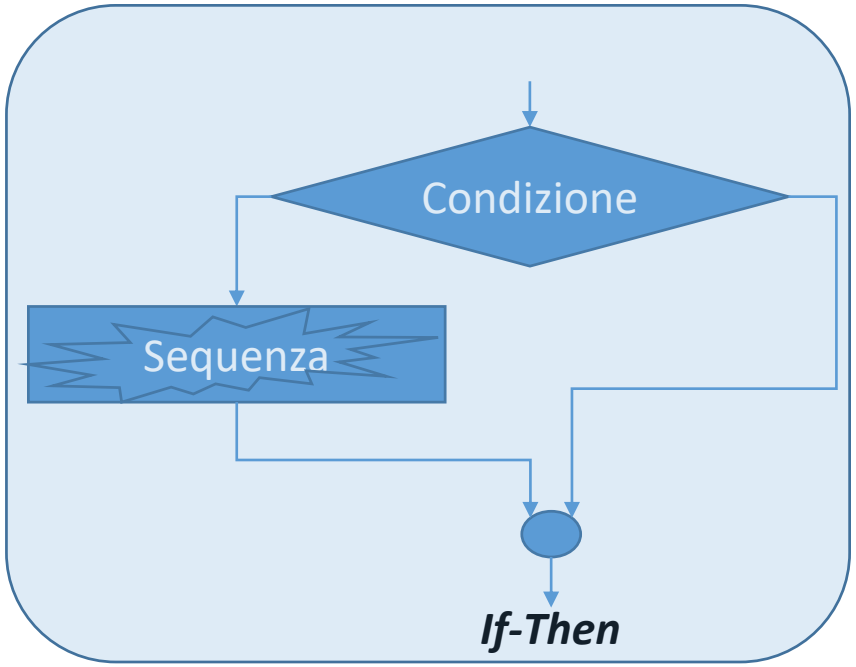


UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso

- I blocchi ***If-Then*** e ***IF-Then-Else*** vengono utilizzati per I costrutti di selezione

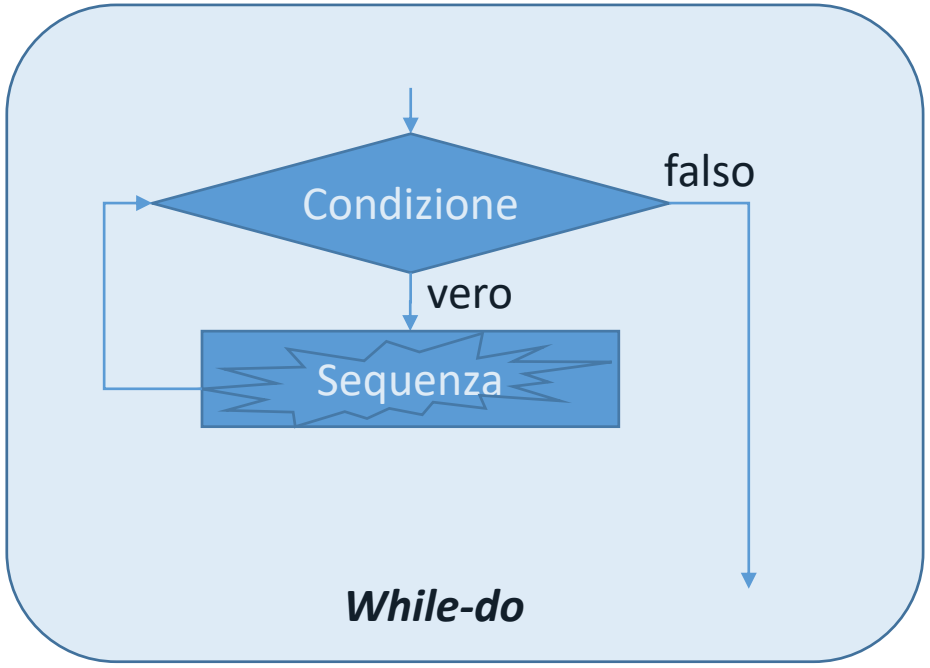


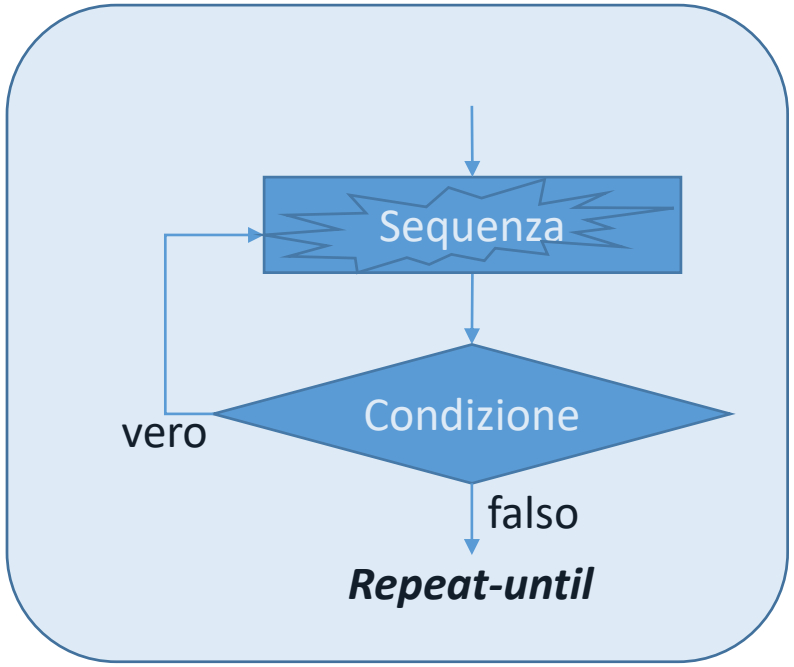


UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso

- I blocchi ***while-do*** e ***repeat-until*** vengono utilizzati per le operazioni cicliche





UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso Strutturati

- Un Diagramma di Flusso si chiama strutturato se:
  - Ha un solo blocco START
  - Ha un solo blocco STOP
  - Utilizza solo sequenze o le strutture avanzate
    - If-then
    - If-then-else
    - While-do
    - Repeat-until

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso Strutturati

Diagramma di Flusso non strutturato

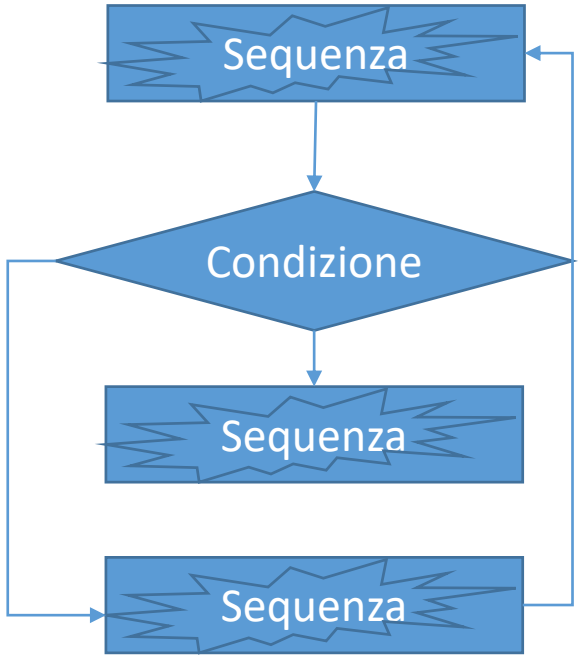
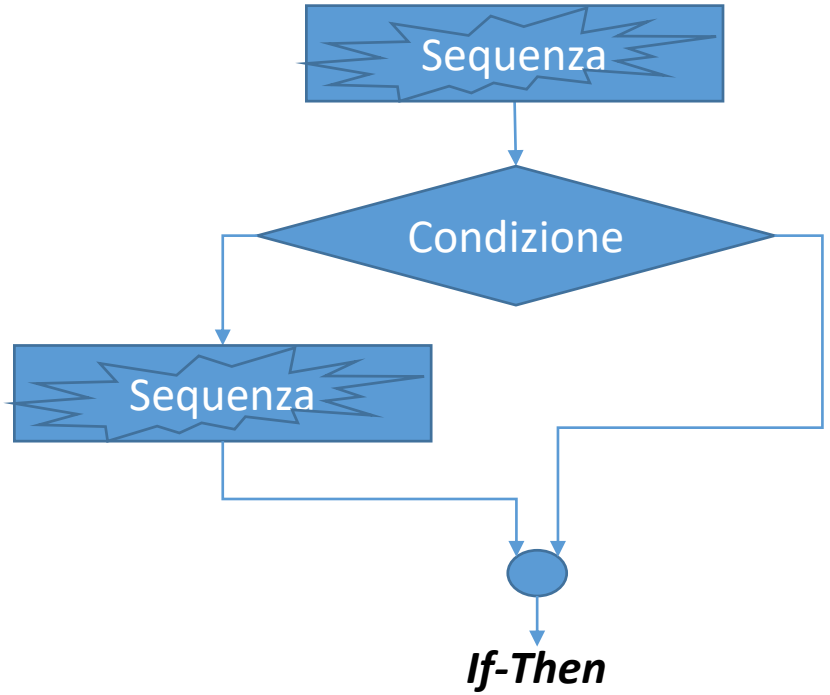


Diagramma di Flusso strutturato



If-Then

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Diagrammi di Flusso Strutturati

Diagramma di Flusso non strutturato

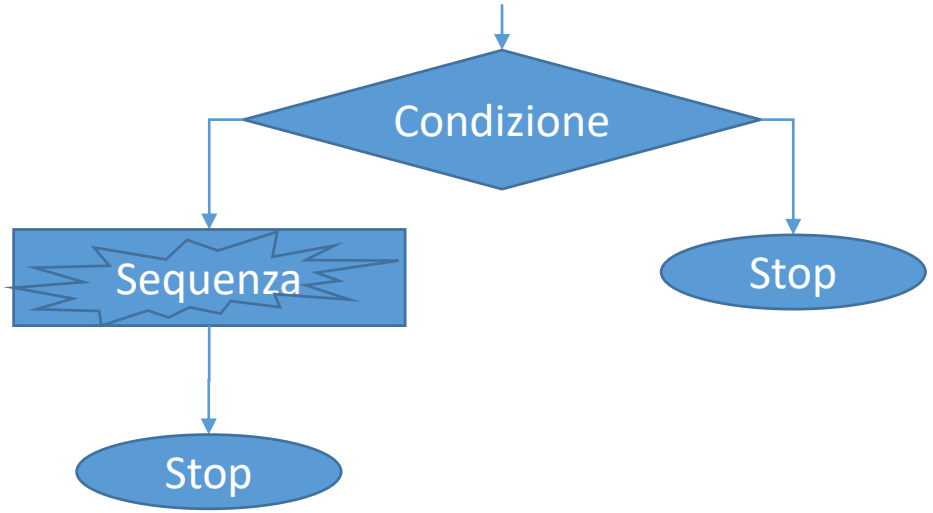
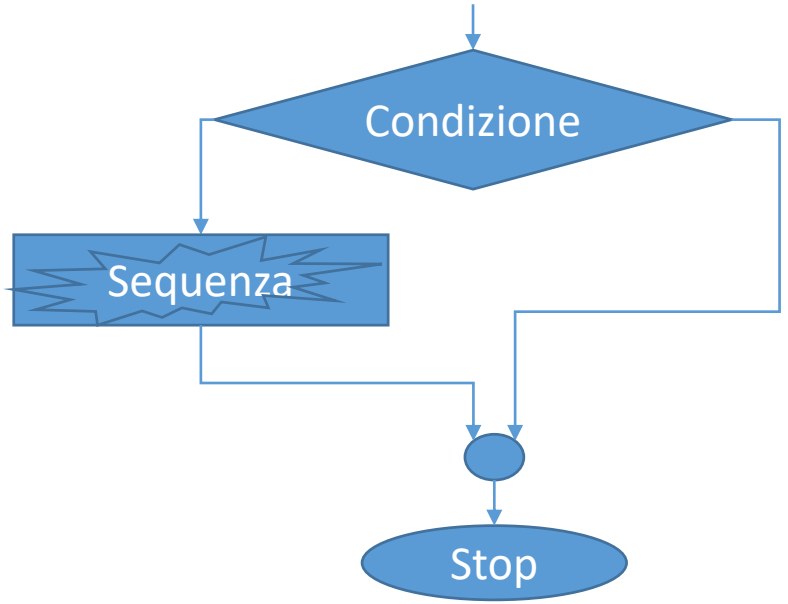


Diagramma di Flusso strutturato



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Teorema di Böhm-Jacopini

Qualunque diagramma di flusso può essere trasformato in un diagramma di flusso strutturato equivalente

- In altre parole è possibile realizzare ogni algoritmo utilizzando solo le strutture avanzate Sequenza, Condizione, Ciclo o iterazione
- I Diagrammi strutturati hanno il vantaggio di essere molto più intuitivi e comprensibili.

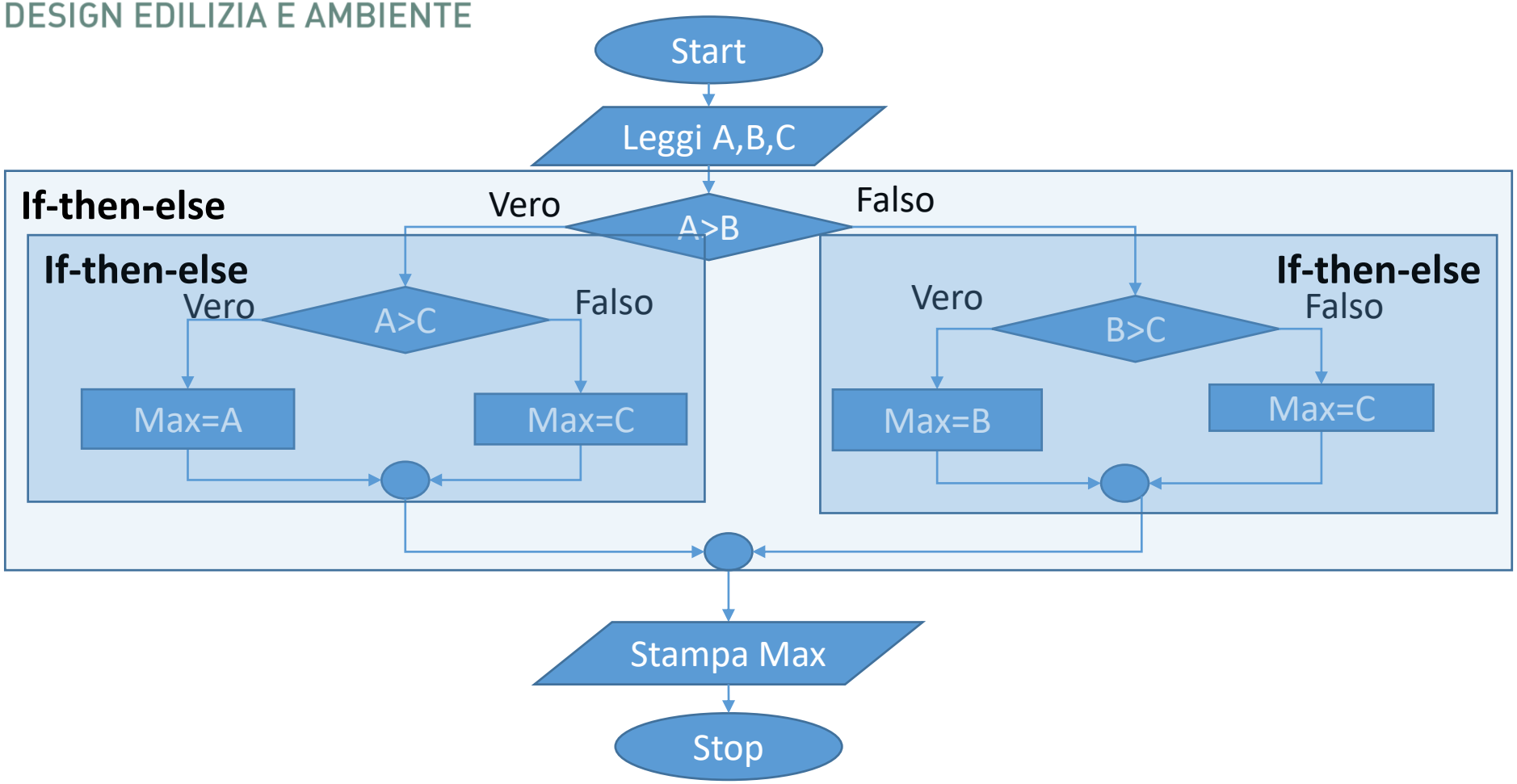
UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 1

Richiedere in ingresso 3 numeri e stampare il massimo

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 1



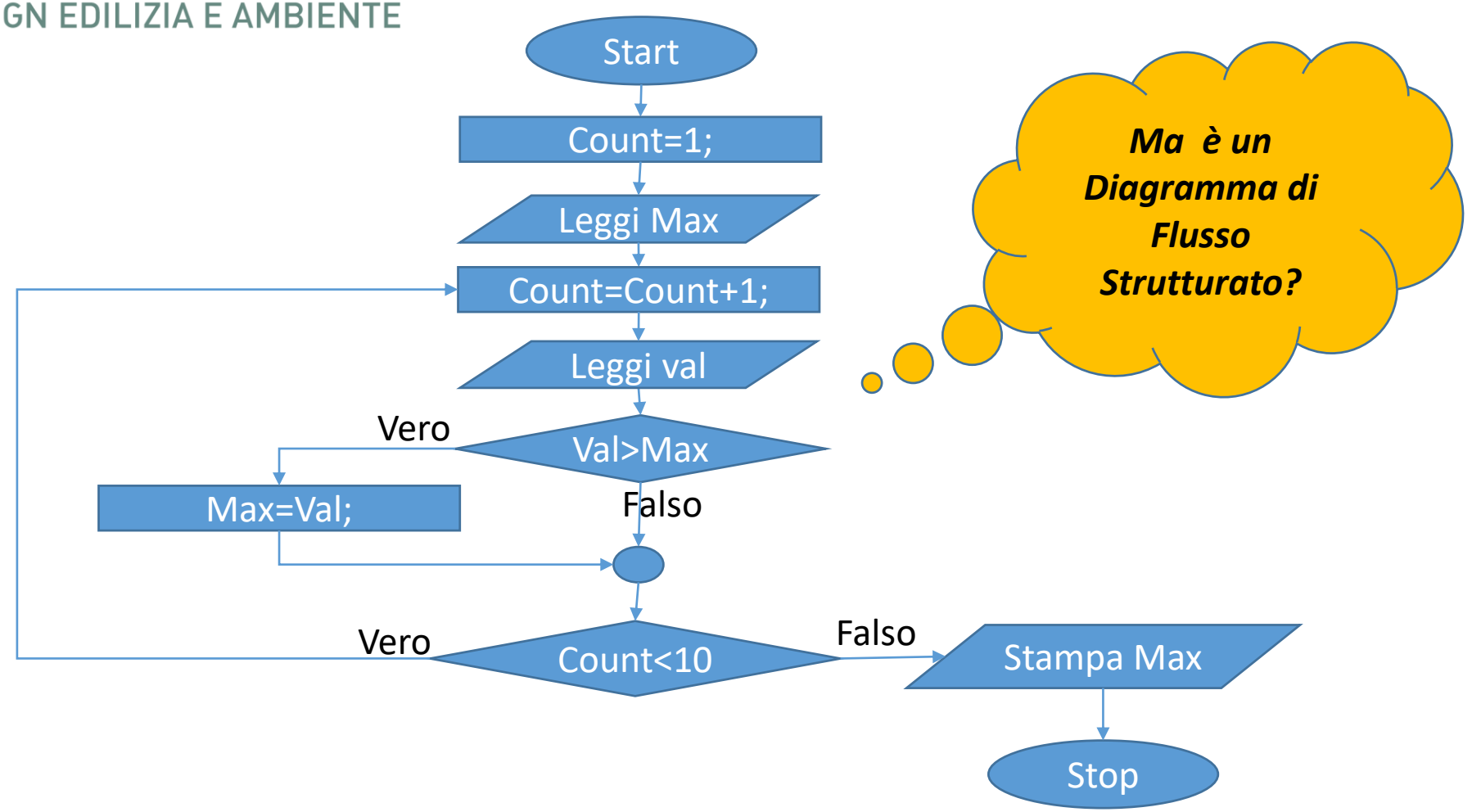
UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 2

Richiedere in ingresso 10 numeri e stampare il massimo

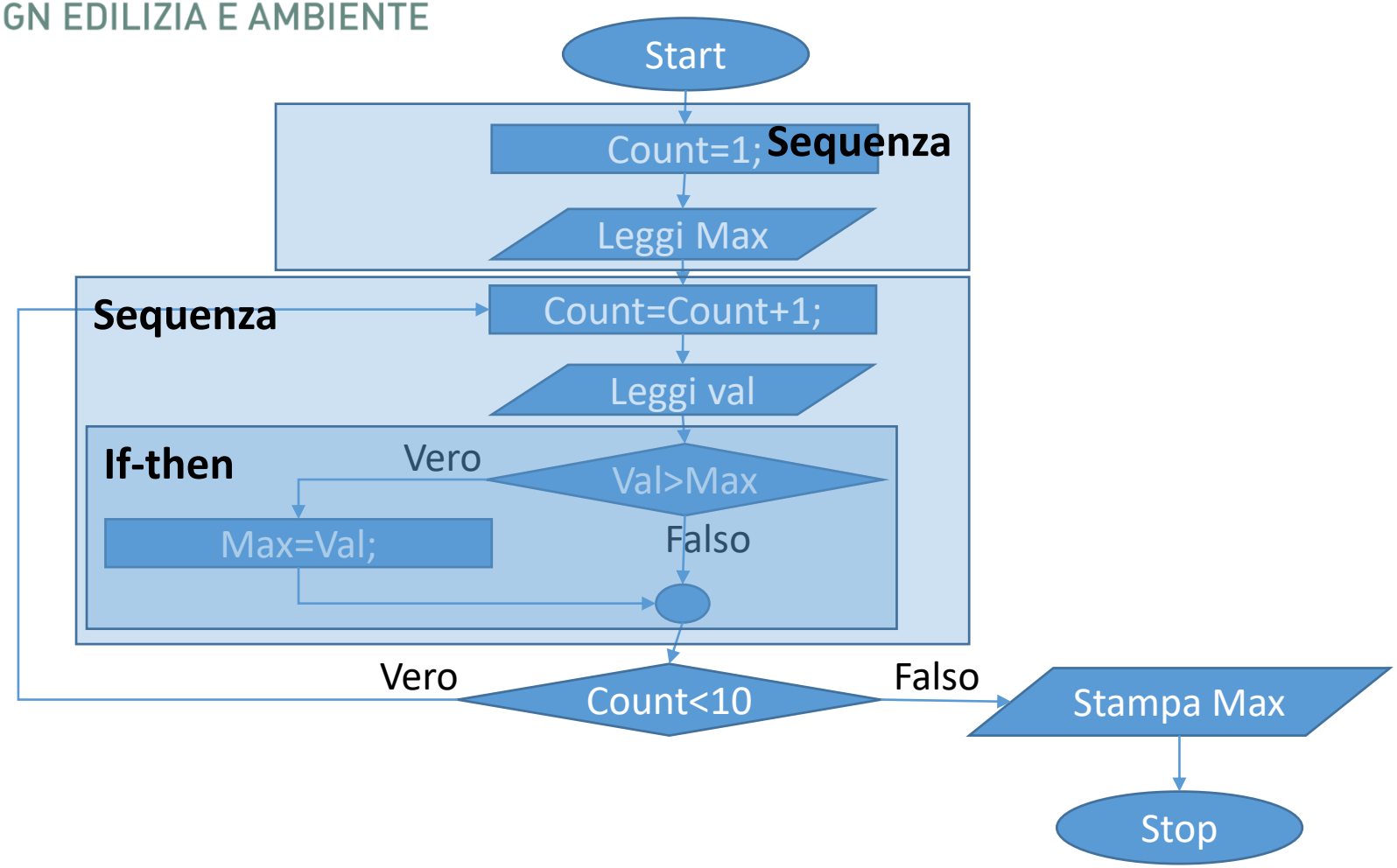
UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 2



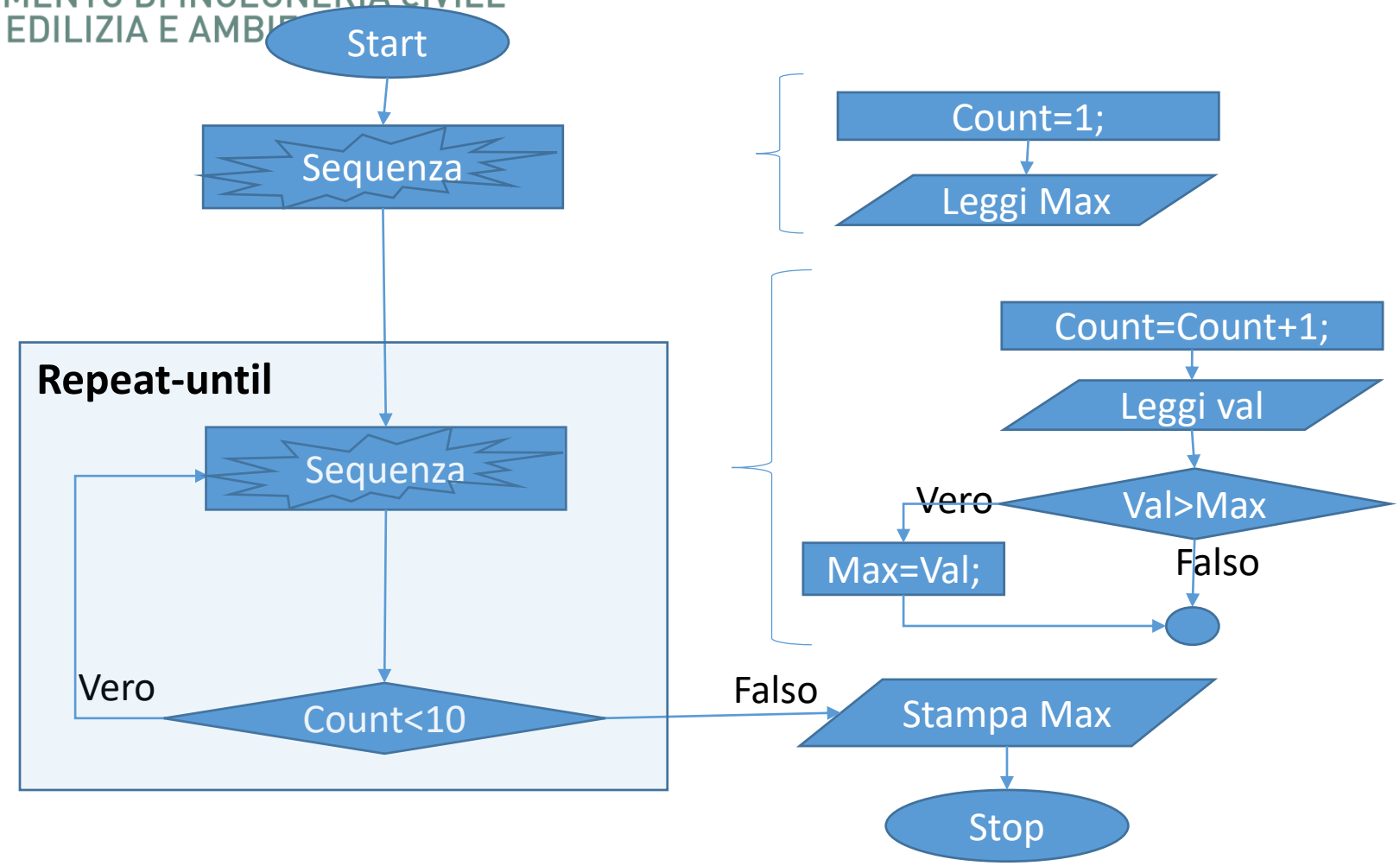
UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 2



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBI

Esempio 2



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 3

Realizzare il Diagramma di Flusso dell'Algoritmo per il calcolo del Massimo Comun Divisore

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 3

Teorema di Euclide:

"ogni divisore comune di a e b è divisore di a, b e del resto r della divisione tra a e b (a *mod* b), se questo non è nullo"

Soluzione di Euclide - Costruzione dell'algoritmo

Dati due numeri ***a***,***b*** assumiamo che **a** sia sempre il maggiore

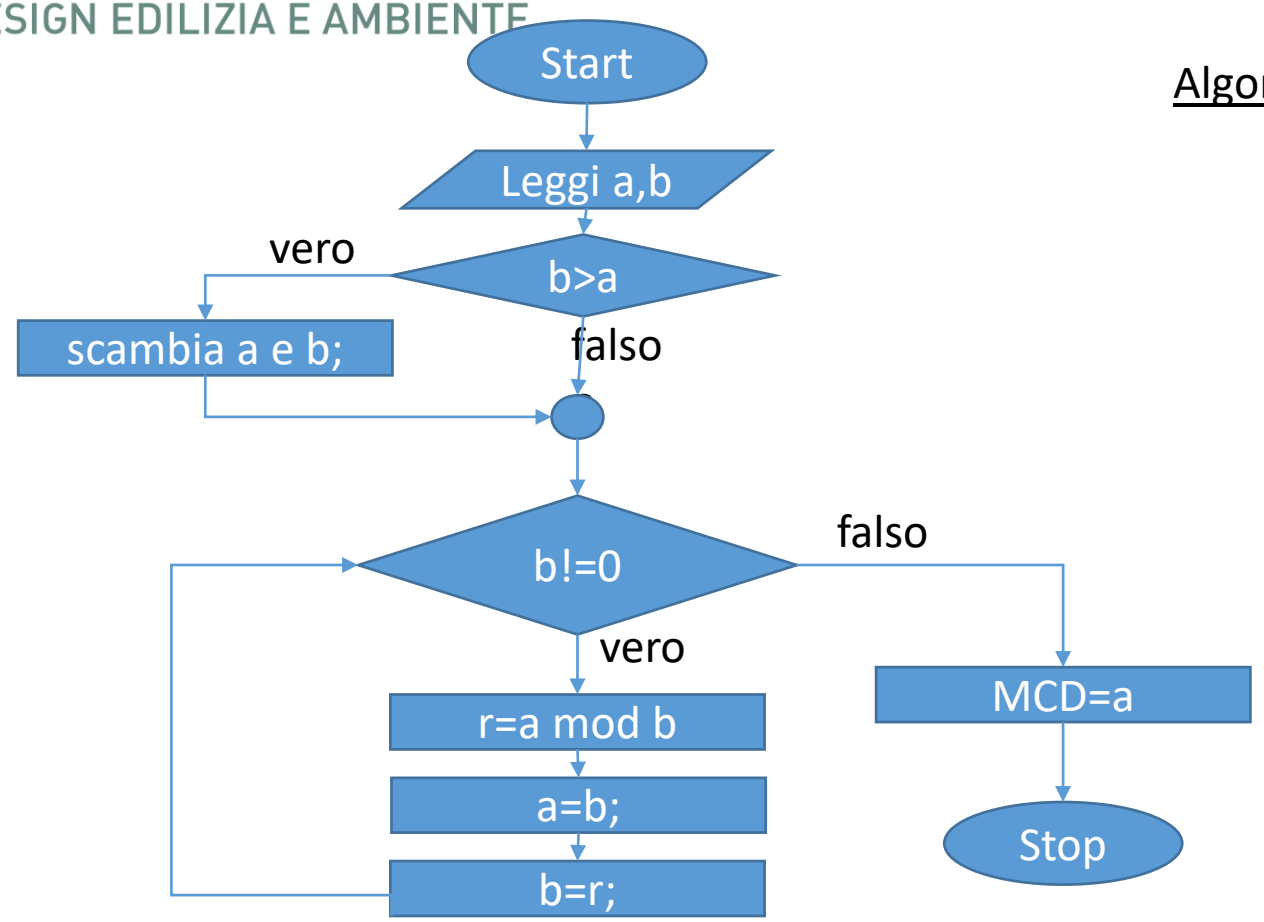
Effettuiamo l'operazione r= a *mod* b se r=0 allor b è MCD di a

Altrimenti calcoliamo r'= b mod r, se r'=0 allora r è divisore di b ed è MCD di a (non lo era b)

Possiamo quindi procedere così fino a trovare il caso in cui l'operazione dia zero.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio 3



Algoritmo:

1. acquisire due numeri a,b
2. se b>a scambiare a con b
3. se b= 0 MCD(a,b)=a e termina
4. r= a *mod* b
5. sostituire a con b, b con r ed andare al passo 3

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Sequenza statica e dinamica di algoritmi

- L'esecuzione di un algoritmo da parte di un esecutore si traduce in una successione di azioni che vengono effettuate nel tempo.
  - Si dice che l'esecuzione di un algoritmo evoca un processo sequenziale, cioè una serie di eventi che occorrono uno dopo l'altro, ciascuno con un inizio e una fine identificabili.
  - Si definisce *sequenza di esecuzione* la descrizione del processo sequenziale. La sequenza di esecuzione è l'elenco di tutte le istruzioni eseguite, nell'ordine di esecuzione, e per questo motivo viene anche detta sequenza dinamica.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio

Ad esempio, nel caso dell'algoritmo di calcolo delle radici di un'equazione di 2° grado viene evocato un unico processo sequenziale, descritto da una sola sequenza di esecuzione che coincide proprio con la descrizione dell'algoritmo.

Calcolo

$$d = \sqrt{b^2 - 4ac}$$

Calcolo la prima radice come

$$x_1 = \frac{-b + d}{2a}$$

Calcolo la seconda radice come

$$x_2 = \frac{-b - d}{2a}$$

Se si modifica l'algoritmo di calcolo delle radici di una equazione di 2° grado per controllare la esistenza di radici reali, si comincia ad osservare come esso genera comportamenti diversi dipendenti dal valore dei dati di input a, b e c.

Calcolo

$$\Delta = b^2 - 4ac$$

Se $\Delta \geq 0$

Allora Calcolo

$$d = \sqrt{\Delta}$$

Calcolo la prima radice come

$$x_1 = \frac{-b + d}{2a}$$

Calcolo la seconda radice come

$$x_2 = \frac{-b - d}{2a}$$

Altrimenti Calcola radici complesse

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Esempio

In questo caso il processo evocato non è più fisso, ma dipende dai dati iniziali da elaborare. Difatti sono possibili le due sequenze di esecuzione

Calcolo

$$\Delta = b^2 - 4ac$$

Verifico che $\Delta \geq 0$ *è risultata vera*

Calcolo

$$d = \sqrt{\Delta}$$

Calcolo la prima radice come

$$x_1 = \frac{-b + d}{2a}$$

Calcolo la seconda radice come

$$x_2 = \frac{-b - d}{2a}$$

Calcolo

$$\Delta = b^2 - 4ac$$

Verifico che $\Delta \geq 0$ *è risultata non vera*

Calcolo radici complesse

A seconda che i valori assegnati ai coefficienti forniscano un discriminante positivo o nullo e negativo. Uno stesso algoritmo ha quindi generato due sequenze di esecuzione tra loro diverse.

Esempio

- Si consideri adesso l'algoritmo seguente che schematizza un gioco che si fa con le 52 carte francesi e che viene detto "solitario del carcerato" per il fatto che non riesce quasi mai o solo dopo molto tempo.

Fintantoché (il mazzo ha ancora carte) ripeti:
Mischia le carte
Prendi 4 carte dal mazzo e disponile sul tavolo di gioco
Se (le 4 carte hanno la stessa figura) allora levale dal tavolo di gioco

- il gioco ha termine quando tutte le carte sono state eliminate.
- L'esempio mostra che un algoritmo può prescrivere più di una sequenza di esecuzione. Se poi l'algoritmo prescrive un processo ciclico, ossia la ripetizione di un insieme di operazioni, allora può accadere che il numero di sequenze sia addirittura infinito. In questo caso l'algoritmo prescrive un processo che non ha mai termine.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Caso Migliore e Caso Peggiore

Fintantoché (il mazzo ha ancora carte) ripeti:
Mischia le carte
Prendi 4 carte dal mazzo e disponile sul tavolo di gioco
Se (le 4 carte hanno la stessa figura) allora levale dal tavolo di gioco

Mischiate le carte
Prese le 4 carte dal mazzo e messe sul tavolo di gioco
Verificato che (le 4 carte hanno la stessa figura) è risultata vera
Tolte le 4 carte dal tavolo di gioco

ripetuto infinite volte
Mischiate le carte
Prese le 4 carte dal mazzo e messe sul tavolo di gioco
Verificato che* (le 4 carte hanno la stessa figura) *è risultata falsa

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Altri Casi

Mischiate le carte
Prese le 4 carte dal mazzo e messe sul tavolo di gioco
Verificato che **(le 4 carte hanno la stessa figura)** ***è risultata falsa***
Mischiate le carte
Prese le 4 carte dal mazzo e messe sul tavolo di gioco
Verificato che **(le 4 carte hanno la stessa figura)** ***è risultata vera***
Tolte le 4 carte dal tavolo di gioco

- Vero al 2 tentativo

- Vero al n-1 tentativo

ripetuto n-1 volte
  Mischiate le carte
  Prese le 4 carte dal mazzo e messe sul tavolo di gioco
  Verificato che **(le 4 carte hanno la stessa figura)** ***è risultata falsa***
Mischiate le carte
Prese le 4 carte dal mazzo e messe sul tavolo di gioco
Verificato che **(le 4 carte hanno la stessa figura)** ***è risultata vera***
Tolte le 4 carte dal tavolo di gioco

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Sequenza Statica e Dinamica

- In un programma
  - …descrizione dell'algoritmo in un linguaggio di programmazione
- la sequenza lessicografica (anche chiamata *sequenza statica*) delle istruzioni
  - …. descrizione delle azioni da svolgere nel linguaggio di programmazione

  descrive una *pluralità di sequenze dinamiche* differenti.
- Il numero di sequenze dinamiche non è noto a priori e dipende dai dati da elaborare.
- La valutazione sia del tipo che del numero delle sequenze dinamiche è di fondamentale importanza per valutare soluzioni diverse dello stesso problema in modo
  - da poter dire quale di esse presenta un tempo di esecuzione migliore
  - da poter affermare se una soluzione ha terminazione o meno.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Linguaggio di Programmazione

- Il linguaggio di programmazione è una **notazione formale** per descrivere algoritmi e, come ogni linguaggio, è dotato di un **alfabeto, un lessico, una sintassi ed una semantica.**
- L'aspetto formale del linguaggio si manifesta soprattutto nella presenza di **regole rigide** per la composizione di programmi a partire dai semplici caratteri dell'alfabeto.
  - L'insieme di queste regole costituisce la **grammatica del linguaggio**.
- Un limitato insieme di regole definisce la **struttura lessicale** del programma
  - .. o unità elementari, cioè le parole del linguaggio.
  - Il lessico, quindi, permette di strutturare l'insieme limitato dei caratteri dell'alfabeto nel vocabolario.
- L'organizzazione delle parole in frasi è invece guidata da regole che compongono la **sintassi del linguaggio**.
- Infine l'attribuzione di un significato alle frasi è oggetto delle **regole semantiche**.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Linguaggio di Alto Livello

- **Linguaggi di Alto Livello**
  - linguaggi di programmazione che, con istruzioni più sintetiche
- *ossia con istruzioni più vicine al tradizionale modo di esprimere i procedimenti di calcolo da parte di un essere umano,*
  - rendono l'attività di programmazione più semplice;
  - Fanno uso di uno *pseudo-linguaggio umano*, utilizzando per il loro scopo *parole-chiave* o *codici operativi* ispirati quasi esclusivamente alla lingua inglese.
- La scrittura dei programmi in linguaggio di alto livello fa in modo che essi non dipendano più dalla CPU.
- E' difatti sufficiente che ciascuna CPU abbia il suo traduttore o interprete per garantire che lo stesso programma sia eseguito da macchine diverse.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Programmazione Strutturata

- Il ruolo degli algoritmi è fondamentale se si pensa che **essi sono indipendenti sia dal linguaggio** in cui sono espressi sia dal **computer che li esegue**.
- Si pensi ad una ricetta per una torta alla frutta:
  - il procedimento è lo stesso indipendentemente dall'idioma usato;
  - la torta prodotta è la stessa indipendentemente dal cuoco.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Progettazione degli Algoritmi

- Il progetto degli algoritmi è una onerosa attività intellettuale
  - …. molto più onerosa di quella di esprimere l'algoritmo con un linguaggio di programmazione
  - che richiede **creatività ed intuito**.
- Non esiste un algoritmo per il progetto degli algoritmi!
- Valutazione della complessità della soluzione individuata:
  - se ci sono algoritmi diversi per descrivere lo stesso processo:
- la *complessità* ci dice quale di essi è migliore, ossia quale viene eseguito nel minor tempo con la minima occupazione di memoria, più in generale con il miglior utilizzo di tutte le risorse disponibili.
- lo studio della *correttezza* ci consente di valutare l'aderenza della soluzione alle specifiche del problema.
  - Essere sicuri della correttezza è un'attività tanto più complessa quanto più complesso risulta il progetto dell'algoritmo.

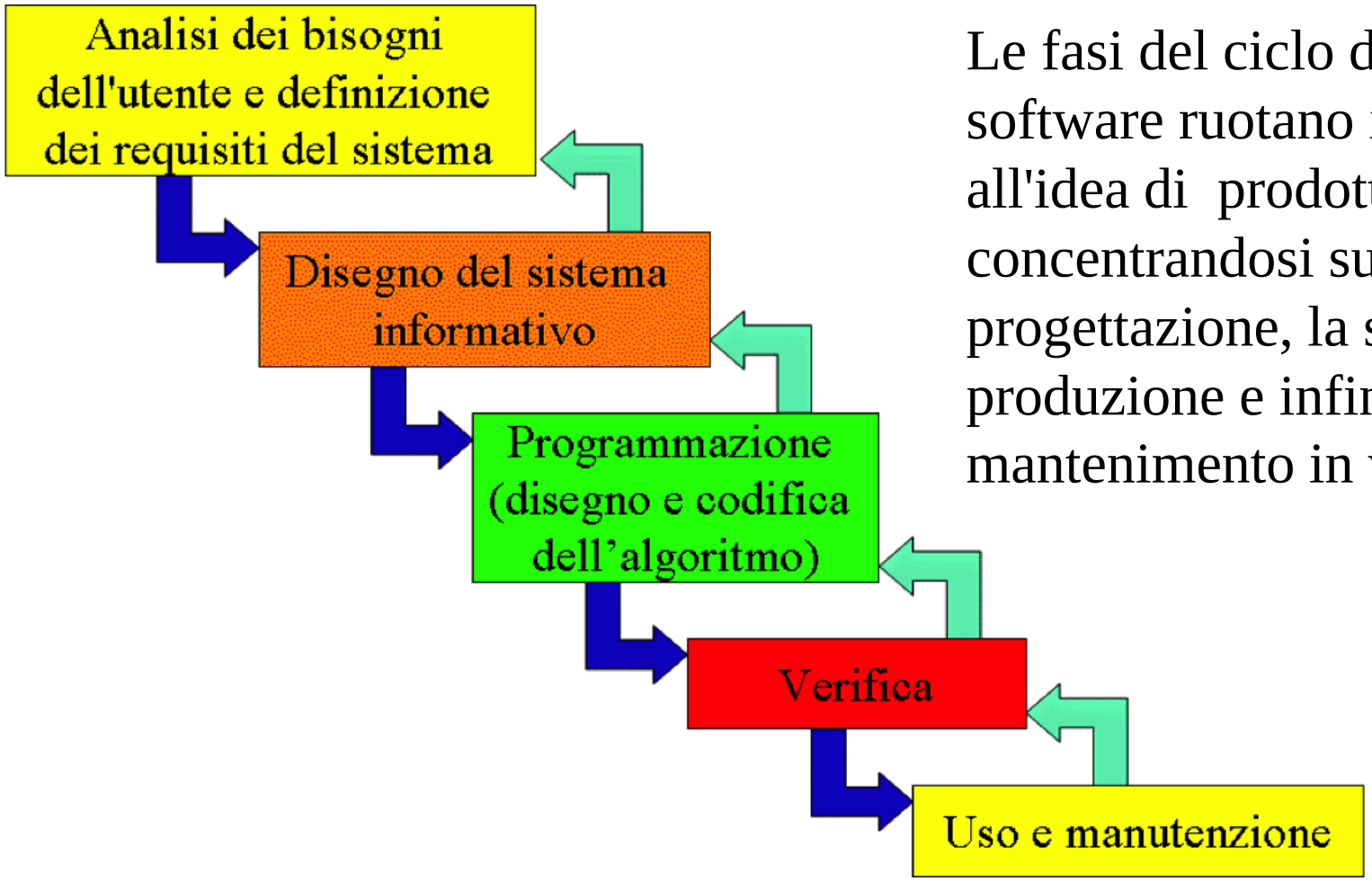
UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Ingegneria del Software

- Ai fini di una produzione industriale di qualità è necessario evitare di produrre software in base alle esperienze e/o alle iniziative del programmatore:
  - il processo di produzione del software non può essere un processo di tipo artigianale
- Ad esempio, negli anni '60 il programmatore usava mille trucchi per risparmiare memoria!
  - deve invece servirsi di **metodologie e tecniche sistematiche** di progettazione e programmazione con fissati parametri di qualità e in maniera standard.
- La "***software engineering***" (ingegneria del software) è la branca dell'Ingegneria Informatica che raccoglie il patrimonio di metodi e tecniche per la produzione del software.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Ciclo di Vita del Software



Le fasi del ciclo di vita del software ruotano intorno all'idea di prodotto concentrandosi sulla sua progettazione, la sua produzione e infine il suo mantenimento in vita.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Tecniche Metodi e Metodologie

- Nell'ambito della software engineering si sono sviluppate tecniche, metodi e metodologie.
- Le **tecniche** mettono a disposizione un linguaggio e degli standard che supportano normalmente alcune fasi del processo di produzione del software.
- I **metodi** consistono in un insieme di tecniche il cui uso é guidato da indicazioni formali.
- Le **metodologie** comprendono un insieme di tecniche il cui uso é guidato da regole procedurali ben precise. Le metodologie, a partire dalla formulazione del problema, consentono:
  - di individuare gli oggetti da elaborare e l'algoritmo;
  - formulare oggetti e algoritmo in modo non ambiguo;
  - progettare oggetti e algoritmi nell'ambiente disponibile.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Programmazione Strutturata: Obiettivo

- L'obiettivo principale della programmazione strutturata consiste nella costruzione di programmi con le seguenti caratteristiche:
  - *leggibilità*: un programma non esaurisce la propria esistenza dopo poche esecuzioni. Pertanto deve essere comprensibile ad altri programmatori;
  - *documentabilità*: un programma deve contenere al suo interno il significato e la motivazione di tutte le scelte di progetto effettuate;
  - *modificabilità*: un programma deve essere strutturato in modo da permettere un rapido adattamento ad una eventuale variazione di alcuni parametri del progetto;
  - *provabilità*: un programma deve essere costruito in modo da facilitare le attività di testing e debugging (controllo della correttezza e correzione degli errori) fin dalle prime fasi del progetto software.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Programmazione Strutturata: Caratteristiche

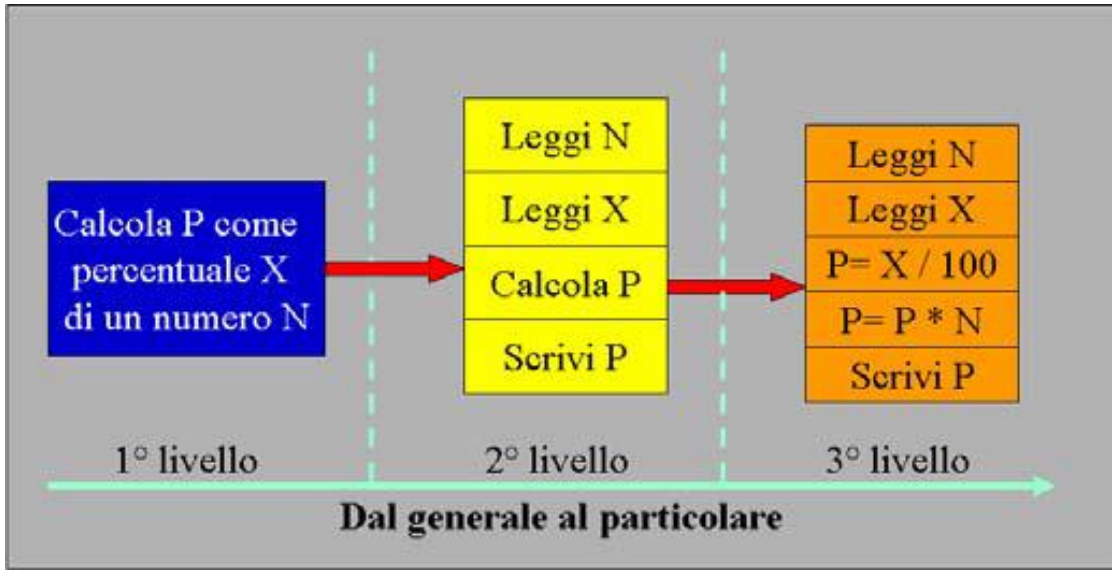
- Le caratteristiche fondamentali della programmazione strutturata sono:
  - **la modularità: ci dice che un programma deve essere composto di moduli funzionali.** Ogni modulo funzionale deve possedere un singolo e ben precisato compito. Ogni modulo inoltre deve essere dotato di un solo ingresso e di una sola uscita.
    - rispecchia la limitazione degli esseri umani ad esaminare un solo aspetto di un problema alla volta.
  - **l'uso di strutture di controllo ad un ingresso e ad una uscita**, che discende dalla necessità che un modulo, composto dall'integrazione di altri moduli tramite le strutture di un controllo, abbia un solo punto di ingresso ed un solo punto di uscita così come i singoli moduli componenti.
  - **l'applicazione del metodo *top-down* o di quello *bottom-up* nella fase di progettazione.**

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Programmazione Strutturata: Top Down

- Il *top-down* e lo *stepwise refinement* costituiscono il modo procedurale di raggiungimento della soluzione. Tale approccio parte dalla considerazione che la complessità di un problema da risolvere non consente di tener conto contemporaneamente di tutte le decisioni realizzative. Sarà quindi necessario procedere per "raffinamenti successivi" procedendo dal generale al particolare.

Ovvero: "*si analizza il problema al più alto livello possibile di astrazione individuandone gli elementi più importanti e supponendo di avere a disposizione un sistema adatto ad eseguire gli elementi funzionali individuati*".

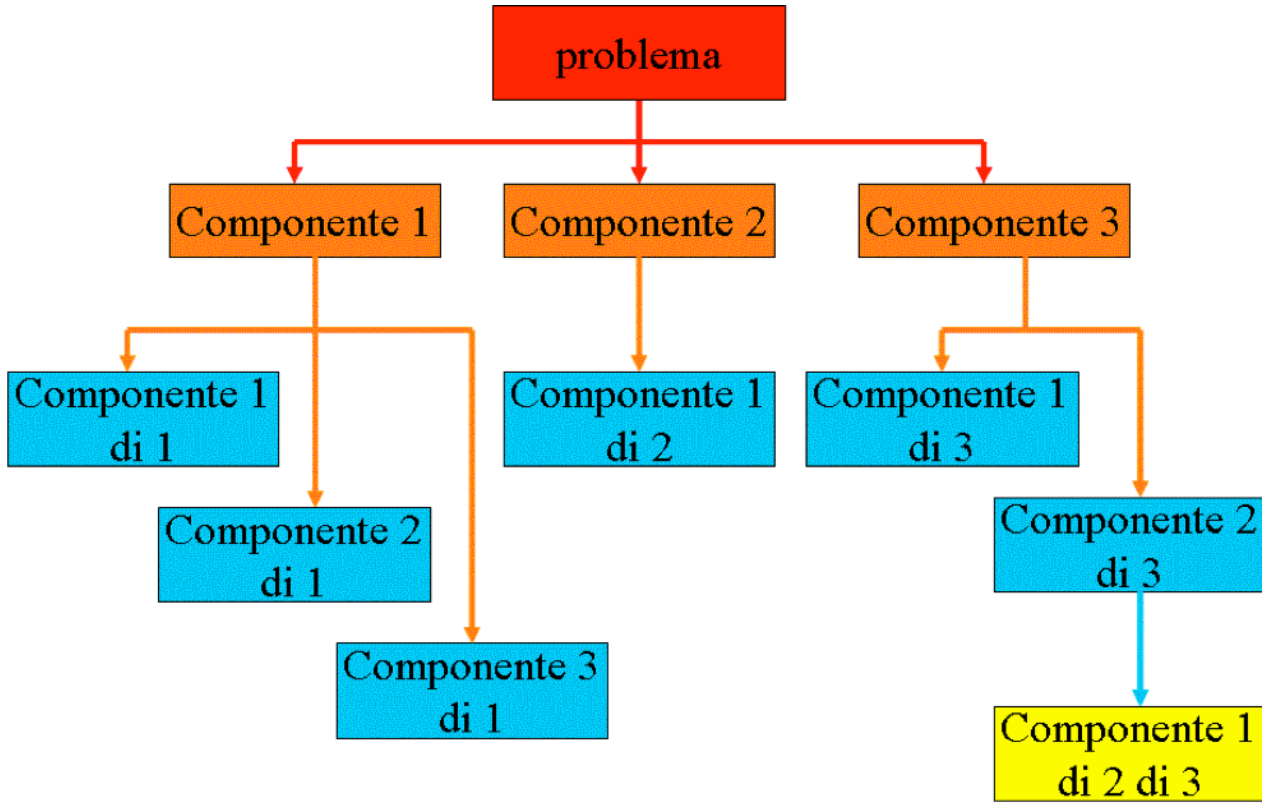


UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Programmazione Strutturata: Top Down

In questo modo la soluzione ad un problema si presenta come un albero in cui :

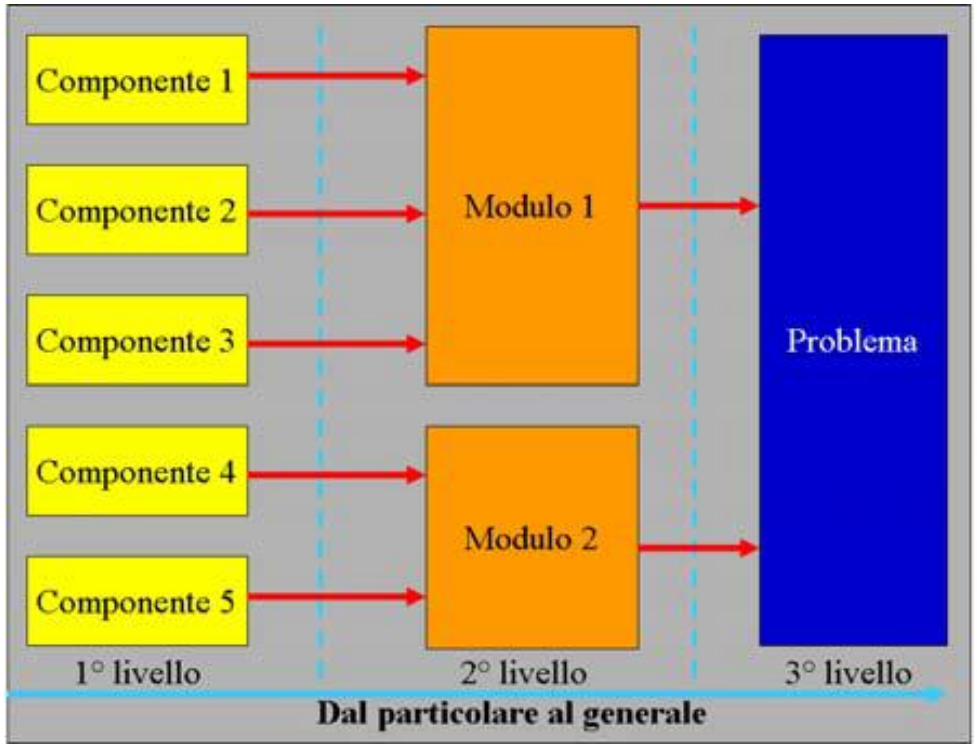
- la radice corrisponde al problema,
- i nodi rappresentano le differenti decisioni di progetto,
- le foglie, infine, vengono associate alla descrizione della soluzione in modo comprensibile all'esecutore.



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Programmazione Strutturata: Bottom Up

Metodo *bottom-up:* parte considerando il sistema reale a disposizione e creando man mano moduli elementari che opportunamente integrati formano moduli capaci di compiere operazioni più complesse. Il procedimento continua fino a quando è stato creato un modulo che corrisponde proprio alla soluzione del problema.



UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Progettazione dei Programmi

- Una delle esigenze maggiormente sentita è quella di una separazione netta, in fase progettuale, tra il “cosa” (analisi dei requisiti e specifiche funzionali) e il “come” (progetto a diversi livelli di dettaglio).
- La distinzione tra il “cosa” e il “come” è comune a qualsiasi tipo di progetto ed è un modo per esprimere con altre parole la separazione fra analisi e sintesi.
- Fasi
  - fase di analisi dei requisiti e delle funzioni
  - fase di progetto
  - analisi critica della soluzione

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Fasi: Analisi

- La fase di analisi dei requisiti e delle funzioni ha lo scopo di definire in maniera non ambigua cosa il programma deve fare.
- fondamentale che le imprecisioni siano eliminate quanto prima per evitare che esse incidano sulle fasi successive del progetto.
  - Infatti i costi di correzione degli errori aumentano quanto più si avanza nel progetto
- ai fini della correttezza occorrerà:
  - verificare che nelle specifiche non vi siano inconsistenze;
  - verificare che le specifiche coprano tutte le possibili situazioni coinvolte nel problema.
- Al termine della fase di analisi si deve disporre della documentazione su:
  - la definizione dei dati di ingresso al problema;
  - la definizione dei dati in uscita dal problema;
  - la descrizione di un metodo risolutivo che sulla carta risolva le specifiche del problema.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Fasi: Progetto

- La fase di progetto può iniziare solo dopo un'accurata fase di definizione dei requisiti e si articola nelle attività di raffinamento successivo dei dati e dell'algoritmo;
  - Inizialmente va scelta la struttura astratta che risponde in maniera più naturale alle caratteristiche del problema.
  - Successivamente, tra tutte le realizzazioni possibili occorrerà scegliere quella che opera il compromesso ottimo dal punto di vista della efficienza e della comprensibilità.
- Per affrontare la scelta in modo adeguato è fondamentale la conoscenza del numero e tipo di operazioni sui dati previste dal problema.
- La tecnica top-down si presenta come un approccio guida con l'obiettivo preciso di ridurre la complessità e di offrire uno strumento di composizione di un'architettura modulare dei programmi.
  - Il modo di procedere di tale approccio è una diretta applicazione del metodo analitico deduttivo che si è rilevato, storicamente, il più adatto alla media degli esseri umani.

Fasi: Progetto

- Procedure per livelli di astrazione. L'astrazione consiste nell'estrazione di dettagli essenziali mentre vengono omessi i dettagli non essenziali.
- Ogni livello della decomposizione presenta una visione astratta dei livelli più bassi in cui i dettagli vengono spinti, quanto più possibile, ai livelli ancora più bassi.
- Il passo di raffinamento iterativo produce, a partire dal problema, una prima decomposizione dell'algoritmo e prosegue con successivi raffinamenti, sempre più vicini all'espressione finale dell'algoritmo nel linguaggio di programmazione.
- Ad ogni passo della decomposizione si devono organizzare i sottoproblemi in:
  - sequenza; quando dall'analisi del problema ci accorgiamo che l'insieme delle attività devono essere svolte una di seguito all'altra;
  - alternativa; quando si deve scegliere tra una o più attività;
  - iterativa; quando una o più attività devono essere eseguite più volte.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Fasi: Analisi Critica

- L’analisi critica è l'ultima fase dell'approccio metodologico.
- Consiste in una minuziosa valutazione della soluzione adottata.
  - Verifica della correttezza della versione dell'algoritmo, mediante ad esempio una simulazione della sua esecuzione
  - Successivamente è possibile compiere una valutazione dell'efficienza della soluzione adottata confrontandola con altre soluzioni oppure studiando l'impatto di una particolare scelta di progetto.
- Ognuna delle azioni descritte deve essere opportunamente documentata in modo che unitamente alle specifiche iniziali si possa avere a disposizione un insieme esaustivo di specifiche utilizzabili per una futura ristrutturazione dell'algoritmo.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Documentazione dei Programmi

- La ***documentazione dei programmi*** è lo strumento fondamentale per la chiarezza e la leggibilità dei programmi. Tali caratteristiche di leggibilità consentono:
  1) una più semplice comprensione di quale sia il problema che il programma risolve e quindi della correttezza del programma stesso;
  2) una più semplice prosecuzione del progetto ogni qualvolta lo si sia interrotto;
  3) una più elementare comunicazione delle scelte di progetto;
  4) una più semplice modificabilità del programma al variare delle specifiche del problema.
- Una buona documentazione deve essere articolata in due livelli:
  a) documentazione esterna del programma;
  b) documentazione interna del programma

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Documentazione Esterna

- La documentazione esterna è il primo livello di documentazione e va compilato preliminarmente nella fase di analisi dei requisiti.
- Costituisce lo strumento fondamentale per l'utente del programma in quanto descrive soltanto il "cosa" fa il programma e non il "come" lo fa.
- La documentazione esterna deve segnalare dettagli operativi tali da rendere autonomo l'utente del programma nell'uso dello stesso. Di essi i più importanti sono:
  a) funzionalità esterne;
  b) attivazione del programma;
  c) diagnostiche di errore;
  d) configurazione richiesta del sistema;
  e) indicazione sull'installazione del programma;
  f) versione e data di aggiornamento.

UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA CIVILE
DESIGN EDILIZIA E AMBIENTE

Documentazione Interna

- La documentazione interna descrive la struttura interna del programma in termini di scelte sulle strutture dati e sull'algoritmo. Tra le principali tecniche a disposizione del progettista per generare la documentazione interna del programma si ricordano:

  1) evidenziazione della struttura del programma mediante l'uso dell'indentazione;

  2) documentazione top-down, ossia facendo comprendere come il programma è stato generato attraverso i vari raffinamenti;

  3) uso di nomi di variabili autoesplicativi, ossia tali da spiegare il ruolo da esse svolte nel corpo del programma;

  4) commento del programma attraverso le frasi di commento di cui tutti i linguaggi di programmazione sono dotati.