

## Unità E1

### Dall'analisi del problema alla definizione dell'algoritmo

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Obiettivi

- Conoscere il concetto di **algoritmo**
- Conoscere le **fasì di sviluppo** di un algoritmo (**analisi, progettazione, verifica**)
- Conoscere i concetti di **risolutore** ed **esecutore**
- Conoscere il concetto di **memoria**
- Essere in grado di **descrivere** semplici algoritmi informali e di seguirne il **flusso** di esecuzione

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Problema


- Problema è un termine largamente utilizzato dai **molti significati** quasi sempre **intuitivi**
- Risolvere un problema
  - Passaggio da uno **stato iniziale**, attraverso **stati intermedi**, ad uno **stato finale**



Qual è la misura dell'ipotenusa di un triangolo rettangolo?

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Non solo problemi matematici



Che fine ha fatto il geometra Emanuele Gargano e il denaro degli abitanti di Vigata? Chi ha ucciso Giacomo Pellegrino [1]?

© 2007 SEI-Società Editrice Internazionale, Apogeo

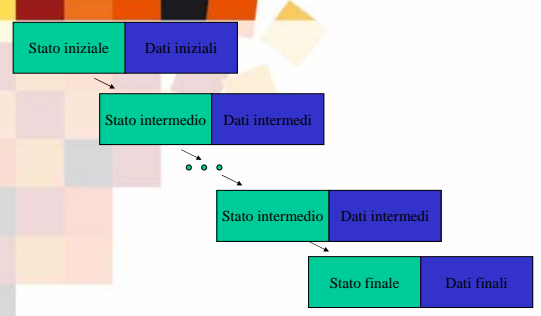
## I dati

- Un certo stato è determinato dai dati disponibili.
- Si parte dunque dai **dati iniziali** per passare a **dati intermedi** e arrivare ai **dati finali** (risultati)

 violino trovato sul luogo del delitto	 impronta digitale rilevata sul violino	 ...	 individuato il colpevole
dati iniziali	dati intermedi	dati intermedi	dati finali
stato iniziale	stato intermedio	stato intermedio	stato finale

© 2007 SEI-Società Editrice Internazionale, Apogeo

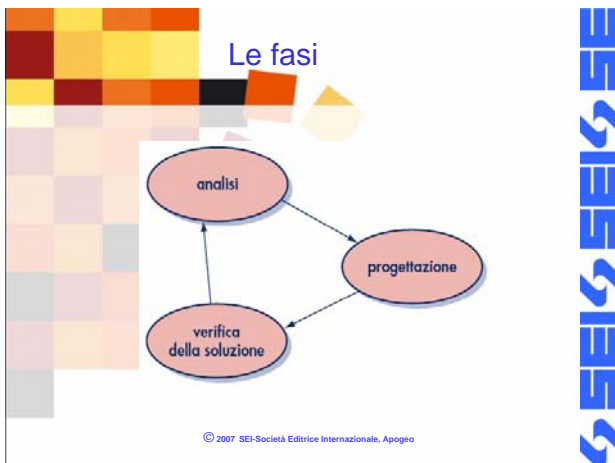
## Risoluzione di un problema



```

graph TD
    A[Stato iniziale] --- B[Dati iniziali]
    B --> C[Stato intermedio]
    C --- D[Dati intermedi]
    D --> E[Stato intermedio]
    E --- F[Dati intermedi]
    F --> G[Stato finale]
    G --- H[Dati finali]
  
```

© 2007 SEI-Società Editrice Internazionale, Apogeo



### Analisi

- Lo **studio** che porta a identificare chiaramente l'**obiettivo** a cui si mira (i risultati a cui si vuol giungere) e di conseguenza i **dati iniziali** significativi che si hanno a disposizione
- Pitagora identifica come obiettivo la ricerca del valore dell'**ipotenusa** di un triangolo rettangolo e come dati iniziali significativi i valori dei due **cateti**.

© 2007 SEI-Società Editrice Internazionale, Apogeo

### Progettazione

- Specifica, attraverso **istruzioni**, le azioni che devono essere intraprese per passare dai *dati iniziali* ai *dati intermedi* ai *risultati finali*
- Le istruzioni date da Pitagora:
  - calcola il quadrato del primo cateto
  - calcola il quadrato del secondo cateto
  - somma i due valori ottenuti
  - calcola la radice quadrata del valore ottenuto

© 2007 SEI-Società Editrice Internazionale, Apogeo

### Verifica della soluzione

- Alla fine deve essere **verificato** che i risultati ottenuti non generino alcuna **contraddizione** con i dati iniziali
- In caso contrario si deve **ripartire** dall'analisi per poi passare di nuovo alla progettazione finché la verifica della soluzione non ha dato esito positivo

© 2007 SEI-Società Editrice Internazionale, Apogeo



## Esecutore




- Deve **comprendere, interpretare correttamente** e deve essere in grado di **eseguire** le istruzioni per trasformarle in azioni
- Il lavoro dell'Esecutore:
  - **eseguire** le istruzioni sui dati iniziali per giungere ai dati finali




© 2007 SEI-Società Editrice Internazionale, Apogeo

## Calcolo dell'ipotenusa



Istruzioni

- prendi due cateti
- calcola il quadrato del primo cateto
- calcola il quadrato del secondo cateto
- somma i due valori ottenuti
- calcola la radice quadrata del valore ottenuto



© 2007 SEI-Società Editrice Internazionale, Apogeo

## Classi di problemi

- Molti problemi hanno radice comune, appartengono alla stessa **classe**
- Uno stesso elenco di istruzioni può servire per la soluzione di problemi specifici che differiscono solo per le informazioni iniziali
- La sequenza di istruzioni dell'esempio precedente ci permette di trovare l'ipotenusa non solo del triangolo con cateti di cm 3 e cm 4 (**problema specifico**) ma di un **qualsiasi** triangolo rettangolo con cateti di dimensione  $x, y$ .
- I cateti  $x, y$  sono i **parametri** che caratterizzano questa classe di problemi
- Da ora in avanti quando useremo il termine problema intenderemo **classi di problemi**

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Il linguaggio

- Il Risolutore deve usare un linguaggio per **comunicare** con l'Esecutore
- Montalbano userà l'italiano o il siciliano con i suoi uomini
- Pitagora ci ha passato la formula con un linguaggio matematico (elevamento a potenza, somma e radice quadrata)

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Algoritmo

- Insieme delle **istruzioni** che permettono ad un esecutore di **partire** dai **dati iniziali** ed **ottenere** i **risultati** desiderati
- Sono **esempi di algoritmi** le procedure che permettono di:
  - effettuare le quattro operazioni matematiche
  - ordinare di una sequenza di numeri
  - verificare la presenza di una parola in un testo
  - simulare il volo di un aereo
  - far diventare il computer un grande giocatore di scacchi.

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Problemi e classi di problemi

- **Problema specifico:** Quali sono le istruzioni per calcolare la somma di  $103+439$ ?
- **Soluzione:** somma 3 con 9, scrivi 2 e porta 1  
somma 3 con 0 e con 1 (il riporto), scrivi 4 e porta 0
- **Problema generale:** Quali sono le istruzioni per calcolare la somma di due numeri interi qualsiasi?
- **Soluzione:** somma la cifra meno significativa del primo addendo con la cifra meno significativa del secondo addendo.....

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Un esempio di problema

- Determinare se un numero è primo
- Analisi:
  - un numero è primo se è divisibile esattamente solo per 1 e per se stesso
  - si cerca il minimo divisore intero maggiore di 1 del numero
  - se è uguale al numero stesso allora questo è primo
- Dato iniziale:
  - un qualsiasi numero intero
- Dato finale:
  - il minimo divisore del dato iniziale maggiore di 1

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Progettazione

- L'idea è quella di provare a dividere il numero per 2, per 3, per 4 e così fino a che il resto della divisione intera è diverso da zero
- I tentativi si esauriscono quando il resto è uguale a zero (si è individuato un divisore esatto del numero)
- Se il divisore è uguale al numero stesso allora si è in presenza di un numero primo

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Algoritmo

1. prendi un numero intero e un suo divisore inizialmente di valore 1
2. aggiungi al divisore il valore 1
3. calcola il resto della divisione intera del numero per il divisore
4. finché il resto è diverso da zero torna all'istruzione b
5. il divisore è il risultato cercato (il numero è divisibile esattamente per il divisore)

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Verifica della soluzione

- Se viene immesso come numero iniziale
  - 35 l'algoritmo restituisce 5
  - 17 restituisce 17 (numero primo)
  - 20 restituisce 2
- E' importante effettuare **prove** di esecuzione dell'algoritmo con **diversi dati** in ingresso
- **Non è possibile dimostrare** però che il procedimento applicato **funziona** per qualsiasi numero intero
- **E' possibile dimostrare** solo che un algoritmo è **errato**, non che è corretto

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Flusso di esecuzione

- L'esecuzione di un'istruzione costituisce un **passo**, la sequenza dei passi il **flusso di esecuzione**
- Un'istruzione in generale può essere eseguita più volte
- Normalmente il numero di passi è diverso dal numero di istruzioni

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Passi - Istruzioni

- | Passi del flusso di esecuzione                          | Istruzioni   |
|---|--|
| 1 avvicinati al bordo del marciapiede                   | a avvicinati al bordo del marciapiede  |
| 2 non ci sono veicoli provenienti da sinistra           | b volgi lo sguardo a sinistra  |
| 3 ci sono veicoli provenienti da destra                 | c volgi lo sguardo a destra  |
| 4 ci sono veicoli in arrivo, torna all'istruzione b     | d finché ci sono veicoli provenienti da sinistra o da destra, torna all'istruzione b |
| 5 ci sono veicoli provenienti da sinistra               | e attraversa la strada   |
| 6 ci sono veicoli provenienti da destra                 |  |
| 7 ci sono veicoli in arrivo, torna all'istruzione b     |  |
| 8 ci sono veicoli provenienti da sinistra               |  |
| 9 non ci sono veicoli provenienti da destra             |  |
| 10 ci sono veicoli in arrivo, vai all'istruzione b      |  |
| 11 non ci sono veicoli provenienti da sinistra          |  |
| 12 non ci sono veicoli provenienti da destra            |  |
| 13 non ci sono veicoli in arrivo (prosegui in sequenza) |  |
| 14 attraversa la strada                                 |  |

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Memoria

- Per poter eseguire le istruzioni che compongono l'algoritmo è necessario poter **memorizzare**
  - i dati iniziali
  - i dati intermedi
  - i risultati finali
  - ma anche le istruzioni stesse
- è necessaria una **memoria**, indipendentemente dal fatto che l'Esecutore sia umano o una macchina

© 2007 SEI-Società Editrice Internazionale, Apogeo

## Automati

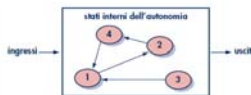
- Un **automa** è una **macchina** in grado di **eseguire** in modo automatico una **sequenza di azioni** predefinite, capace di **acquisire** elementi dall'esterno e **restituire** elementi all'esterno.



© 2007 SEI-Società Editrice Internazionale, Apogeo

## Stato interno

- In un certo istante di tempo la macchina ha una sua condizione interna, detta stato, che può cambiare durante il funzionamento passando a un altro stato



© 2007 SEI-Società Editrice Internazionale, Apogeo

## Macchina di Turing

- Una Macchina di Turing è composta da:
  - un nastro di carta prolungabile a piacere e suddiviso in celle vuote o contenenti al più un unico carattere;
  - un'unità o organo di controllo costituita da uno stato interno, da un insieme di regole e dall'interprete delle regole stesse;
  - una Testina di Lettura/Scrittura (TLS) per leggere e scrivere;



© 2007 SEI-Società Editrice Internazionale, Apogeo

## Tesi di Church

- “qualsiasi funzione calcolabile attraverso un qualsivoglia procedimento meccanico può essere calcolata utilizzando un'opportuna macchina di Turing”



© 2007 SEI-Società Editrice Internazionale, Apogeo

## Proprietà degli algoritmi

- gli algoritmi hanno un numero finito di istruzioni;
- le istruzioni devono essere eseguibili da un esecutore;
- ogni istruzione deve essere eseguita in un tempo finito;
- l'esecutore ha una memoria;
- il calcolo avviene per passi discreti;
- non c'è un limite alla lunghezza dei dati di ingresso;
- non c'è un limite finito alla quantità di memoria;
- sono ammesse esecuzioni con un numero di passi illimitato;
- il calcolo non può essere probabilistico;
- sono ammesse esecuzioni con un numero di passi infinito.

© 2007 SEI-Società Editrice Internazionale, Apogeo



## Sintesi

- La **soluzione** di un problema è costituita dalle istruzioni che applicate ai dati iniziali restituiscono i dati finali (risultati)
- Il **risolutore** svolge le attività di analisi, progettazione e verifica per definire l'algoritmo
- Una **classe di problemi** è l'insieme dei problemi che differiscono solo per i dati iniziali.
- L'esecutore **esegue le istruzioni** sui dati iniziali e giunge ai risultati cercati.
- L'esecuzione di un'istruzione costituisce un **passo**, la sequenza dei passi il **flusso di esecuzione**.
- Le istruzioni vengono eseguite nella sequenza in cui si presentano a meno che un'istruzione non modifichi il flusso di esecuzione.
- La **memoria** è necessaria all'esecutore, umano o macchina, per memorizzare i dati e le istruzioni stesse.

© 2007 SEI-Società Editrice Internazionale, Apogeo

