Laboratorio di Informatica T (Ch1)

Informatica

Cos'è l'informatica?

Che cos'è l'Informatica?

Cos'è l'informatica?

Che cos'è l'Informatica?

Non è facile da definire!

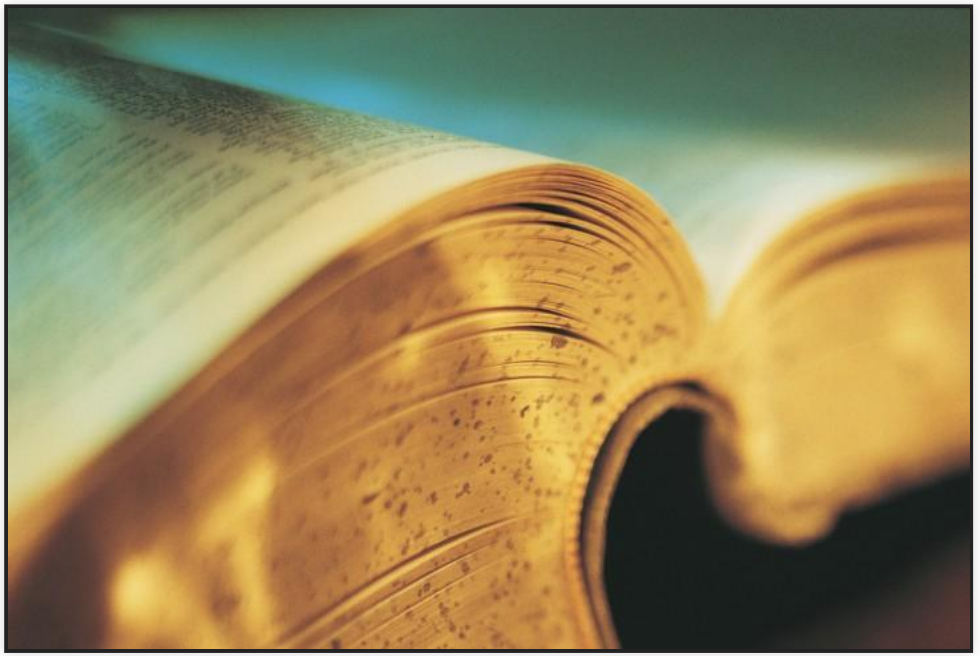
Alcune affermazioni vere:

- L'informatica è parente stretta della matematica
- Ha a che fare con il modo in cui risolviamo i problemi
- Si può fare anche senza un calcolatore!

Vediamo qualche esempio…

Un Primo Esempio

Problema: cercare un parola su un dizionario



Come risolverlo?

Un Primo Esempio

Proviamo a descrivere un metodo di soluzione:

- Sia w la parola da cercare
- Aprire il dizionario a caso
- Siano w', w'' le parola in cima alla pagina sx/dx
- Se $w < w'$, ci restringiamo alle pagine precedenti e ripetiamo
- Se $w > w''$, ci restringiamo alle pagine successive e ripetiamo
- Altrimenti cerchiamo w sulla pagina

Un Secondo Esempio

Problema: trovare il massimo di un insieme di tessere numerate

Come risolverlo?

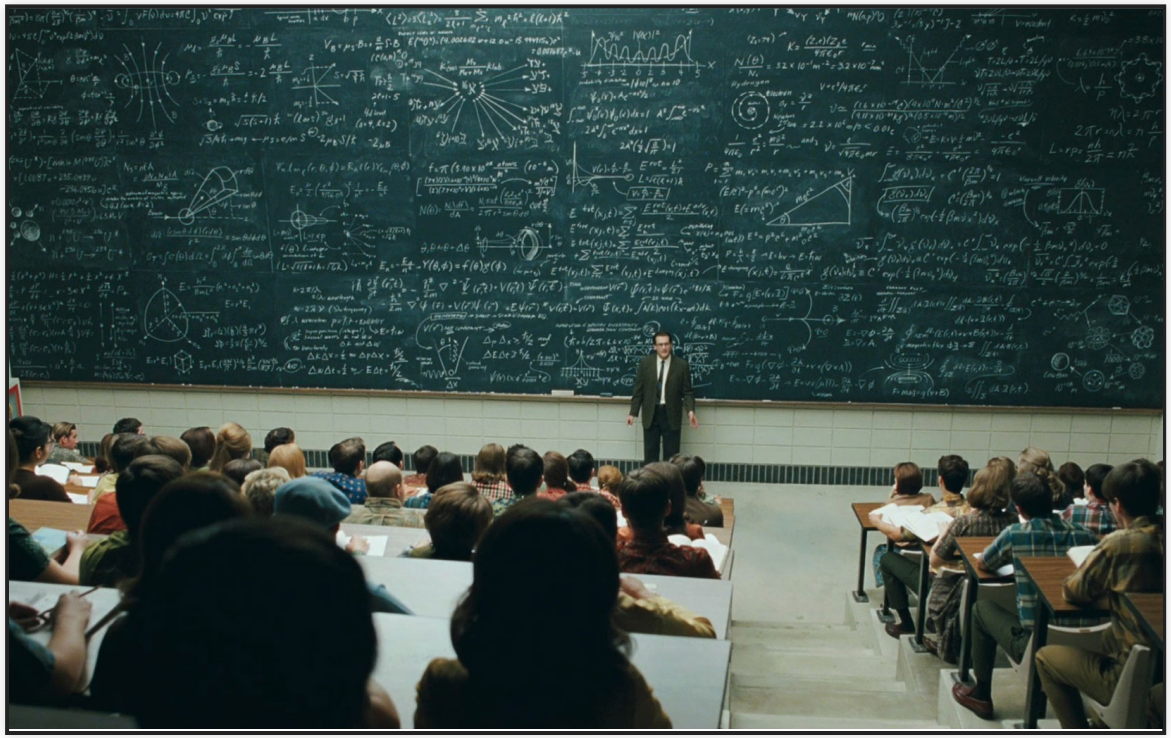
Un Secondo Esempio

Proviamo a descrivere un metodo di soluzione:

- Prendiamo in mano un numero a caso (sia questo m)
- m è temporaneamente il nostro massimo
- Prendiamo in mano tutti gli altri numeri uno per volta
- Sia v il numero corrente
- Se $v > m$, allora v è il nuovo massimo. Mettiamo da parte m
- Altrimenti, mettiamo da parte v e passiamo al prossimo numero

Un Terzo Esempio

Problema: riposizionarci per ordine alfabetico



Come risolverlo?

Un Terzo Esempio

Proviamo a descrivere un metodo di soluzione:

- La prima coppia è in ordine?
- Se non lo è, si scambia
- Poi guardiamo la seconda coppia e così via
- Alla fine, si ripete tutto il processo
- Quando non ci sono più scambi, l'aula è ordinata

Algoritmi e Programmi

Algoritmi

Quelli che abbiamo visto sono esempi di algoritmi

Un algoritmo è processo che risolve un problema

- È un po' generico, ma cattura l'idea fondamentale
- Proviamo ad essere più formali…

Sia data una funzione:

$$f : D_I \mapsto D_O$$

- D_I, D_O sono il dominio di ingresso e di uscita (input/output)
- f descrive quali input vanno mappati in quali output

In altre parole: una funzione descrive un problema

Algoritmi

Se una funzione definisce un problema, allora…

Un algoritmo è un procedimento che computa (o valuta) una funzione

$$\mathbf{f : D_I \mapsto D_O}$$

Una funzione può ammettere diversi algoritmi

- In tal caso, gli algoritmi di dicono equivalenti
  - Per uno stesso valore in D_I…
  - …Due algoritmi equivalenti producono lo stesso valore in D_O
- In questo corso ne vedremo tonnellate!
  - Ognuno di voi risolverà gli stessi problemi in modo diverso

Proprietà Fondamentali degli Algoritmi

Un algoritmo deve soddisfare alcune proprietà fondamentali

Ce ne sono diverse, ma a noi ne interessano due in particolare:

Eseguibilità: Ogni azione elementare dell'algoritmo deve essere eseguibile (in tempo finito)

Non Ambiguità: Ogni azione deve essere interpretabile in modo non ambiguo

- Vedremo ora alcune conseguenze di queste proprietà

Eseguibilità => Elaboratore

Per eseguire un algoritmo è necessario un elaboratore

Questo è un elaboratore:

Eseguibilità => Elaboratore

Per eseguire un algoritmo è necessario un elaboratore

Questo è un elaboratore:

Eseguibilità => Elaboratore

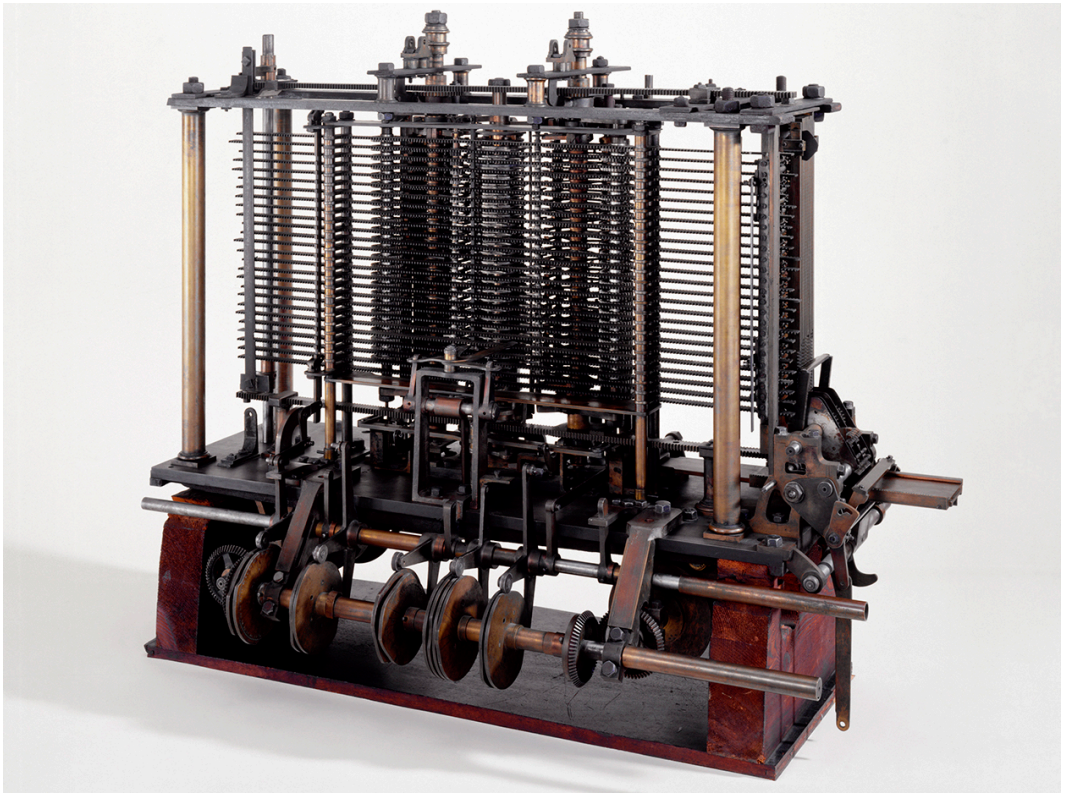
Per eseguire un algoritmo è necessario un elaboratore

Questo è un elaboratore (Wozniak, Apple I, 1976):

Eseguibilità => Elaboratore

Per eseguire un algoritmo è necessario un elaboratore

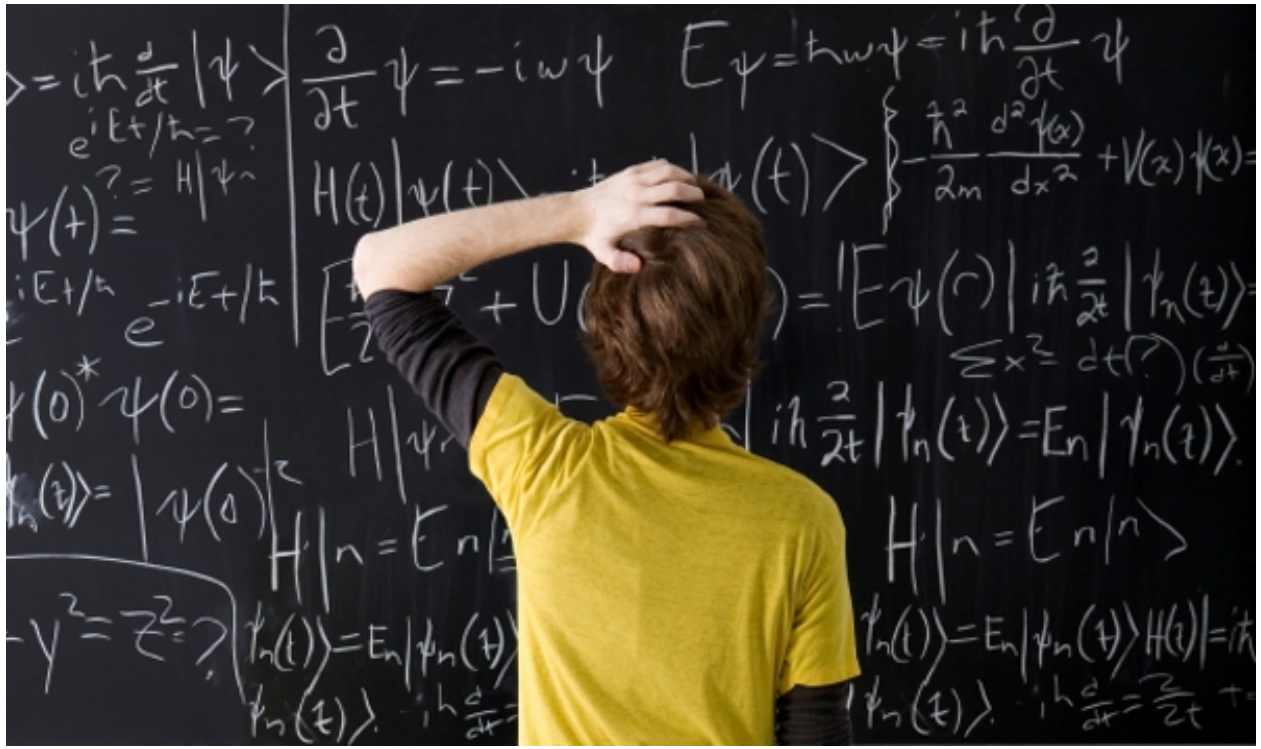
Questo è (parte di) un elaboratore (Analytical Engine, 1837):



Eseguibilità => Elaboratore

Per eseguire un algoritmo è necessario un elaboratore

Questo è un elaboratore (beh, soddisfa la definizione):



Cos'è un Elaboratore

Dal nostro punto di vista:

Un elaboratore è una entità che può:

- Memorizzare informazioni
- Eseguire sul tali informazioni alcune operazioni elementari

Quindi l'elaboratore determina:

- I dati elementari che possiamo usare
- Le operazioni elementari che possiamo effettuare

E.g. dati: numeri interi, operazioni: '+', '<', '='

Cos'è un Elaboratore

Dal nostro punto di vista:

Un elaboratore è una entità che può:

- Memorizzare informazioni
- Eseguire sul tali informazioni alcune operazioni elementari

Una osservazione importante:

- Per poter eseguire qualunque algoritmo…
- …Sono sufficienti pochissimi tipi di dato ed operazioni elementari

Dimostrazione dovuta ad Alan Turing, Alonzo Church (negli anni '30)

Non-ambiguità e Formalizzazione

Consideriamo ora la non-ambiguità:

Per definire in modo non ambiguo un algoritmo:

Ci serve un linguaggio (per esempio testuale) che:

- Deve avere una sintassi non-ambigua
  - Cioè: delle regole sintattiche rigide e ben definite
- Il linguaggio deve avere una semantica non-ambigua
  - Cioè: ogni componente del testo ha un solo significato

Un linguaggio di questo tipo si chiama linguaggio formale

Linguaggi di Programmazione

Si chiama linguaggio di programmazione un linguaggio formale:

- La cui semantica è definita in base ad operazioni elementari…
- …Che possono essere eseguite su un elaboratore

> **Si chiama programma un testo scritto in un linguaggio di programmazione**

- Quindi, un programma è un testo che può essere eseguito!
- **Un algoritmo può essere definito usando un programma**
- Spesso si usa il termine “codificare” o “implementare”

Linguaggi di Programmazione - Un Esempio

Questo è un programma per l'esempio 2 (max di una serie di numeri)

```
V = randi(20000, 1, 2000);
m = V(1);
for v = V(2:end)
    if m < v
        m = v;
    end
end
```

- È scritto in Matlab, il sistema che utilizzeremo
- Non cercate di capire i dettagli adesso: ci arriveremo!

Informatica: la Scienza degli Algoritmi?

L'informatica è la scienza degli algoritmi?

Non esattamente: **non tutti i programmi sono algoritmi**

E.g. Programmi che non "calcolano un risultato"

- E.g. server web, interfacce utente…
- Questi programmi usano algoritmi, ma non sono algoritmi

Ma in sostanza cosa fa un programma?

- Rappresenta informazioni
- Elabora informazioni

Il che ci porta alla definizione di Informatica che adotteremo!

L'Informatica è la scienza della rappresentazione e dell'elaborazione dell'informazione