


# Computabilità

Paolo Baldan

Università di Padova
Corso di Laurea Magistrale in Informatica


# Gzip ...

- Siamo nel 2150. Il Prof. Mat, matematico, fa sempre più fatica a tener traccia dei numeri di telefono dei suoi amici ...
- Con la scoperta di nuovi pianeti abitati, i numeri sono sempre più lunghi ...
- Come memorizzarli in modo “efficiente”?


# Gzip ...

- Prof. Hal, un informatico, suggerisce: "*Nella rubrica, invece che il numero scrivi un programma che lo generi!*"
- Es.: Invece che

```
12345678901234567890123456789012345678
```

scrivi

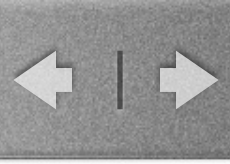
```
for i=1 to 38 do print (i mod 10)
```


# Gzip ...

- Conviene sempre?
- No! Ci sono numeri n tali che, per ogni programma P che genera n

$$size(n) \leq size(P)$$

$\Rightarrow$ numeri casuali

# Gzip ...

- Mat osserva "*Ci sono infiniti numeri casuali!*"
- (Perché?)


# Gzip ...

- Mat: "*Ma Hal, potresti comunque aiutarmi scrivendo un programma che determini se un numero è casuale?*"
- Hal risponde "*No!*"
- Ma non è maleducato ...


# Gzip ...

... il programma richiesto non esiste!!!


# Esercizio

- Provare che
  - I numeri casuali sono infiniti
  - Non esiste un programma che determina se un numero è casuale


# Non tutto si calcola!

- Ci sono dei problemi non risolubili da un calcolatore ...
- E non sono problemi strampalati
  - Halting problem
  - Correttezza dei programmi
  - .....


# Obiettivo del corso

- Potenza e limitazioni dei sistemi di calcolo
- Domanda:
  "quali problemi possono si possono risolvere con un computer / con una procedura effettiva?"
- Limitazioni teoriche intrinseche (indipendenti dal particolare modello di calcolo)


# Domande specifiche

- Cos'è una procedura effettiva?
- Cosa vuol dire che un problema è risolto da una procedura effettiva?
- Caratterizzazione dei problemi risolvibili e non risolvibili
- Relazioni tra i problemi non risolvibili (grado di difficoltà)

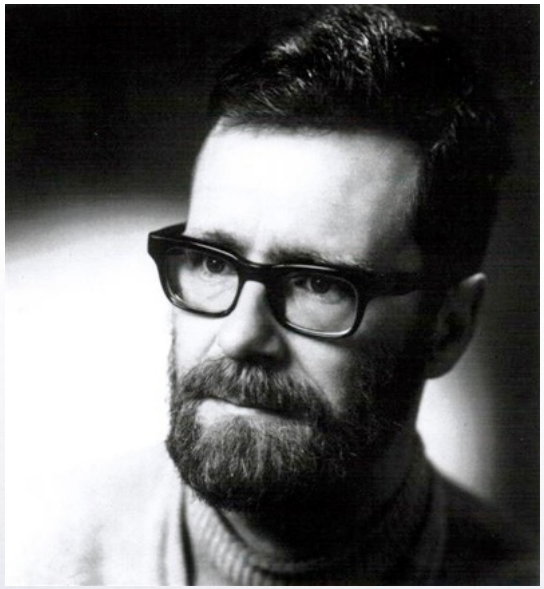

# E le risorse?

- Qui consideriamo i problemi risolvibili/non risolvibili senza limitazioni di risorse (memoria e tempo)
- Teoria della complessità
  - Si occupa delle risorse
  - Classifica i problemi risolvibili in una gerarchia di "difficoltà"


# Un po' di storia ...

- Quando l'informatica inizia a preoccuparsi di questioni di computabilità?
- Prima di nascere!
- L'informatica moderna nasce dalla scoperta dei suoi limiti, da un grande fallimento ...

# Cos'è l'informatica?



*"Computer science is no more about computers than astronomy is about telescopes."*

unsourced: Edgser Wjbe Dijkstra
[Rotterdam 1930 - Nuenen 2002]

Premio Turing 1972, persona non facile...
la citazione potrebbe benissimo esser sua

"A Case against the GO TO Statement"
letter to Communications of ACM 3, 1968

http://www.cs.utexas.edu/users/EWD/


# Cos'è l'informatica?

*"Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes, or chemistry is about beakers and test tubes. Science is not about tools. It is about how we use them, and what we find out when we do."*

M.R. Fellows & I. Parberry, "SIGACT Trying to Get Children Excited About CS", Computing Research News 5(1), 1993


# Cos'è l'informatica?

*"Computer science is the study of algorithms, including*

*their formal and mathematical properties*
*their hardware realizations*
*their linguistic realizations*
*their application"*

N.E. Gibbs & A.B. Tucker, "A model curriculum for a liberal arts degree in CS", Communications of ACM 29(3), 1986 [citato da M.B. Schneider& J.L. Gersting, An invitation to Computer Science 2nd edition, 1998]


# Cos'è l'informatica?

- La capacità di costruire e usare degli strumenti, secondo una procedura (codificata), è carattere distintivo dell'uomo
- Homo habilis o homus informaticus?

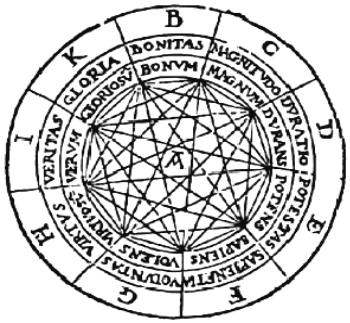
# Una storia: Sbagliando s'impara!

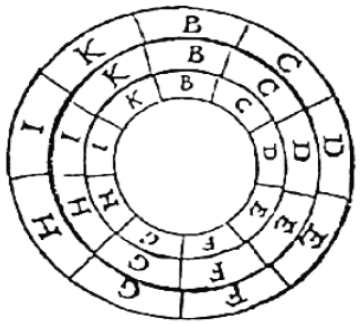
- Logica come studio dei meccanismi del ragionamento (capacità di trarre conseguenze da insiemi di premesse).
- Schemi di ragionamento indipendenti dal "senso" delle singole parti del discorso
  - Sillogismo aristotelico: Ogni uomo è mortale, Socrate è uomo quindi...
- idea di un procedimento generale su base combinatoria per trovare tutte le verità
- inscindibile legame con la storia dell'informatica

# Lullus

- Raimundus Lullus (Palma di Maiorca 1232 - 1315)
  - Insegna a Parigi e Montpellier; dopo una crisi mistica, frate francescano
  - Su *Ars Magna* (1305) divulga un mezzo meccanico (ruota lulliana, progettato nel 1275) per combinare concetti e nozioni
    - Concetti, entità, domande, virtù, vizi -> lettere B-K
    - Ruote concentriche per combinarle
    - Ottenuta la combinazione si può riflettere sul significato
  - Metodo per confutare le tesi di ebrei e musulmani, e convertirli

Tavola 1 - Il modello delle rotule lulliane e la tavola delle dignità



# Il sogno di Leibniz

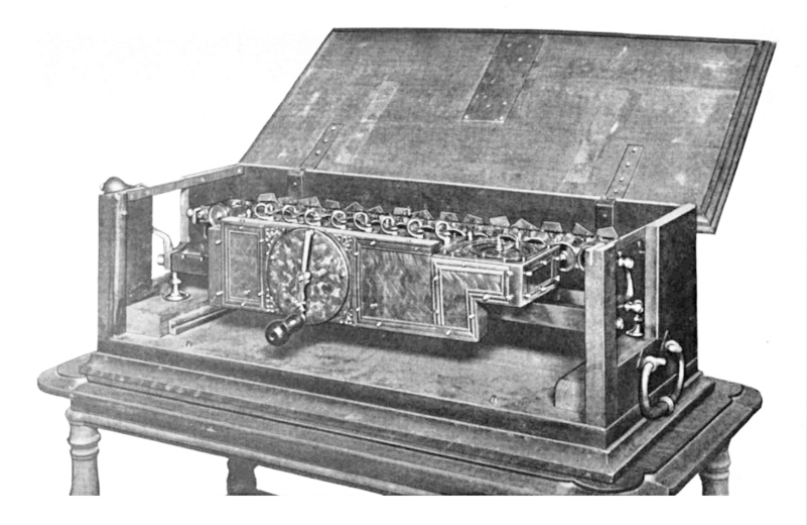
- Gottfried Wilhelm Leibniz (Lipsia 1646 - Hannover1716)
  - "ideare una certa scrittura o una lingua atta a rappresentare perfettamente le relazioni tra i nostri pensieri"
  - lingua "filosofica" universale per facilitare apprendimento e sviluppo del sapere
    - Caracteristica Universalis: lingua universale per esprimere ogni nozione e concetto
      - Sistema di simboli e nozioni di base (enciclopedia universale)
      - Grammatica ideale per combinarli (latino semplificato)
    - "Calculus ratiocinator": ragionamento attraverso manipolazione formale di simboli
      - dai ragionamenti su Dio alla matematica

# Calculemus!

- "Quando emergeranno delle controversie tra filosofi, non occorreranno discussioni, così come non ne occorrono tra due contabili. Basterà infatti prendere carta e penna e dire "Calcoliamo!"
- Non solo visionarietà
  - Si scomponga il concetto di uomo in animale razionale e se ne considerino i componenti come i termini primi. Si assegni ad es. ad animale il numero 2 e a razionale il numero 3. Il concetto di uomo sarà esprimile come 2x3 ovvero 6.
  - Affinché una proposizione sia vera occorre che, esprimendo in misura frazionale il rapporto soggetto-predicato (S/P) [...] il numero del soggetto sia esattamente divisibile per il numero del predicato.
  - Ad es. la proposizione "tutti gli uomini sono animali" si riduce alla frazione 6/2 ottenendo come risultato un numero intero (3). La proposizione è dunque vera.

# Dalla teoria alla pratica

- Non solo visionarietà ("theoria cum praxis")
  - Prima macchina calcolatrice [la *Staffelwalze*, 1674] per computare le quattro operazioni aritmetiche [contro le due della *Pascaline*], dove prodotto e divisione sono ricondotte a somma e sottrazione.



forse dopo lo *Speeding Clock* del 1623, opera di Wilhelm Schickard (Herrenberg 1592 - Tubinga 1635)

# Algebra di Boole

- George Boole (Lincoln 1815 - Ballintemple 1864)
  - Con Leibniz le regole del ragionamento iniziano ad essere espresse per mezzo di un formalismo simbolico
  - Con Boole la logica viene interpretata come un'algebra
    - Nelle asserzioni "Ogni pianta è vivente" e "Ogni uomo è mortale" piante, viventi, uomini, mortali sono visti come classi
    - L'algebra booleana fornisce le leggi di tali classi
  - Ad esempio...
    - x = animali che sono pecore, y = entità bianche, x*y = pecore bianche !!

# Algebra di Boole

- Leggi sulle classi, con in particolare (come in Leibniz)
  - x * x = x
- Algebra della logica = algebra ordinaria ristretta a 0, I (interpretati come la classe vuota e universale, rispettivamente)
- Interpretazione per + e -
  - x + y = entità che stanno nella classe x o in y [idem per x - y]
- Con semplici manipolazioni algebriche...
  - x * x = x ---> x - x * x = 0 ---> x * (I – x) = 0

...“principio di non contraddizione” della Metafisica di Aristotele (è impossibile che una qualità appartenga e non appartenga alla stessa entità).

# detour

- Charles Babbage (Londra 1791 - Marylebone 1871)
  - Non discuteremo di Babbage (né di Aiken e degli Harvard Mark I-IV)
  - Difference Engine (n.II: progetto 1849; costruzione 1991) è una macchina *special purpose* per il calcolo di polinomi che usa il metodo delle differenze finite [da f(1) derivo il valore di f(1.1) etc.], non potendo moltiplicare...
  - Analytical Engine (dal 1837 in poi) è una macchina *general purpose* programmabile (schede perforate), con memoria e unità aritmetica (*mill*)
    - mai prodotta, sostanzialmente dimenticata fino ai 1930s, ci ha lasciato il nome di un linguaggio (ADA, da Ada King, contessa di Lovelace) e buona fantascienza...

# Frege

- Gottlob Frege (Wismar 1848 - Bad Kleinen 1925)
  - La logica di Boole rappresentava un ramo della matematica che comprendeva il ragionamento logico... circolarità!!
  - Frege poneva la logica come fondamento di tutta la matematica
    - Ideografia (*Begriffsschrift,* 1879: calcolo formale dei simboli, 100p): linguaggio formale artificiale con regole sintattiche e di manipolazione (logica dei predicati)
      - variabili & quantificazione
        - tutti i cavalli sono mammiferi: $\forall x.\text{cavallo}(x) \Rightarrow \text{mammifero}(x)$
        - alcuni cavalli sono purosangue: $\exists x.\ \text{cavallo}(x) \wedge \text{purosangue}(x)$
  - Lettera di Russel (1902): interessante, ma... inconsistente! [$\{ x \mid x \notin x \}$]


In questo corso non useremo slide

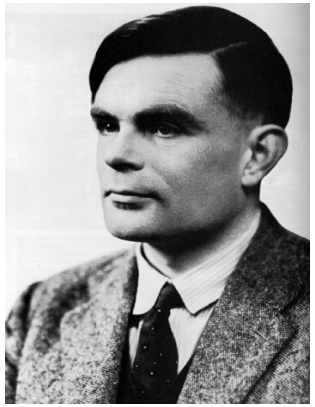
# Hilbert

- David Hilbert (Wehlau 1862 - Gottinga 1943)
  - Traduzione formale del sogno di Leibniz, spogliato da ambizioni metafisiche
    - Matematica (a partire dalla geometria: *Grundlagen der Geometrie*, 1899) ridotta ad un sistema formale (assiomi + regole)
    - Consistenza (come fondamento per l'esistenza, opposto al costruttivismo) da dimostrare con metodi finitistici
  - Entscheidungsproblem (EP, "problema della decisione"), esprimibile come
    - "dato un sistema completo [per la logica del primo ordine], trovare un algoritmo di decisione [che dica se una formula è provabile]"

# Godel rompe tutto!

- Kurt Gödel (Brno 1906 - Princeton1978)
  - Colpo terribile al programma logicista di Hilbert
  - Teoremi di incompletezza dell'aritmetica (1931)
    - Primo: in ogni teoria T assiomatizzabile e consistente della aritmetica elementare esistono enunciati F per i quali non si può provare "F" e neppure "non F" (intuitivamente, non si riesce a provare né la falsità né la verità di F)
    - Secondo: Nelle stesse ipotesi, in T non si può provare la consistenza di T stessa.

# Turing fa il resto



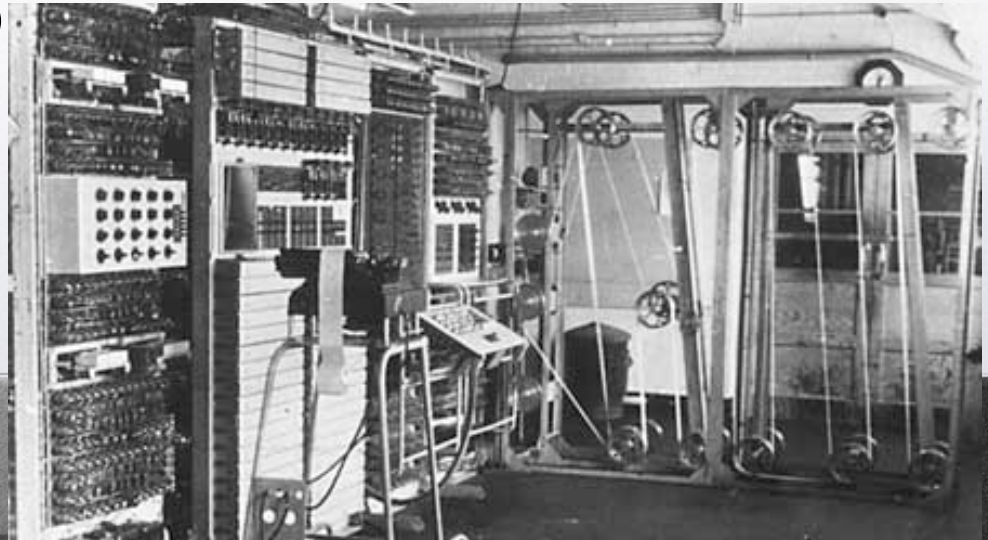
- Alan Turing (Londra 1912 - Wilmslow 1954)
  - Macchina di Turing (MdT)
    - Macchina che legge/scrive simboli su di un nastro in base ad un "programma": a partire da dati di input produce un output
  - Formalizzazione della nozione di procedimento di calcolo
    - Algoritmo: concetto antichissimo...
      - Euclide (Grecia, 300-400 A.C.): Algoritmo per il MCD
      - Al-Chwarizmi (Persia, 800 D.C.): Algoritmi per le operazioni in notazione decimale

# Turing (cont.)

- MdT
  - Nozione solida di funzione calcolabile
    - MdT Calcola tutto ciò che è calcolabile (Tesi di Church-Turing)
  - Non può calcolare tutto
    - Halting problem: Dati programma P e input x, l'esecuzione di P termina su x?
  - Deduzioni (hilbertiane) come funzioni calcolabili -> soluzione negativa di EP
- Esistenza di una MdT universale [macchina *general purpose*]
  - accetta come input dati e la descrizione di una MdT [programma], espressi nello stesso linguaggio, ed esegue il programma sui dati
  - Univ(x,y) esegue il programma x sul dato y

# Turing (cont.)

- Lavora nel 1943 a Bletchey Park al Colossus (calcolatore *special purpose* per decifrare i codici nazisti) e intuisce la potenzialità dell'elettronica
- Contribuisce al progetto del primo calcolatore programmabile
  - Mai realizzato perché ritenuto oltre i limiti tecnologici (6k di memoria!!!)
  - Misconosciuto per la necessità di mantenere il segreto di stato (l'esistenza del Colossus rivelata nei 1970s)



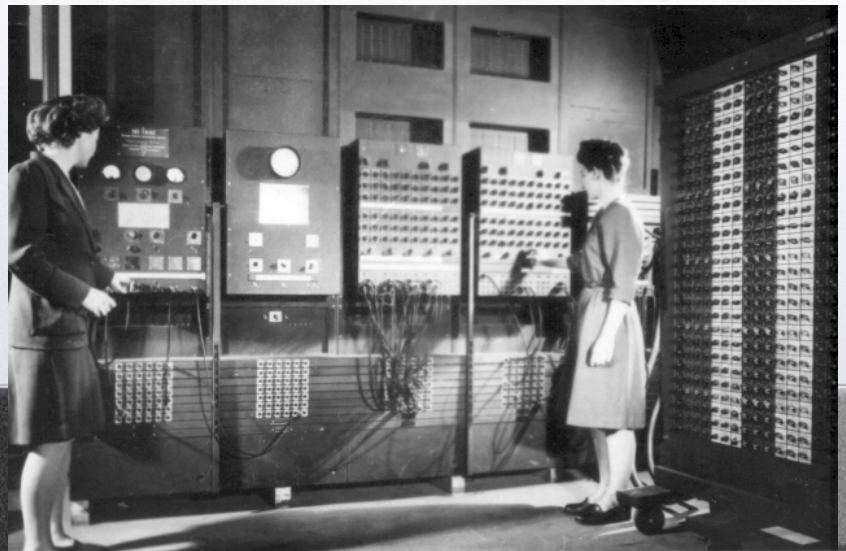
Colossus X

# Il calcolatore moderno

- John Von (János) Neumann (Budapest 1903 - Washington 1957)
  - Dall'idea MdT Universale (programma e dati in memoria...) nasce il computer digitale programmabile moderno
  - EDVAC (Electronic Discrete Variable Automatic Computer) Report del 1945 ricordato come l'origine del computer
  - [Turing era molto avanti, e dovremmo citare Konrad Zuse (Berlino 1922 - Hünfield 1992), ed il suo Z3...]

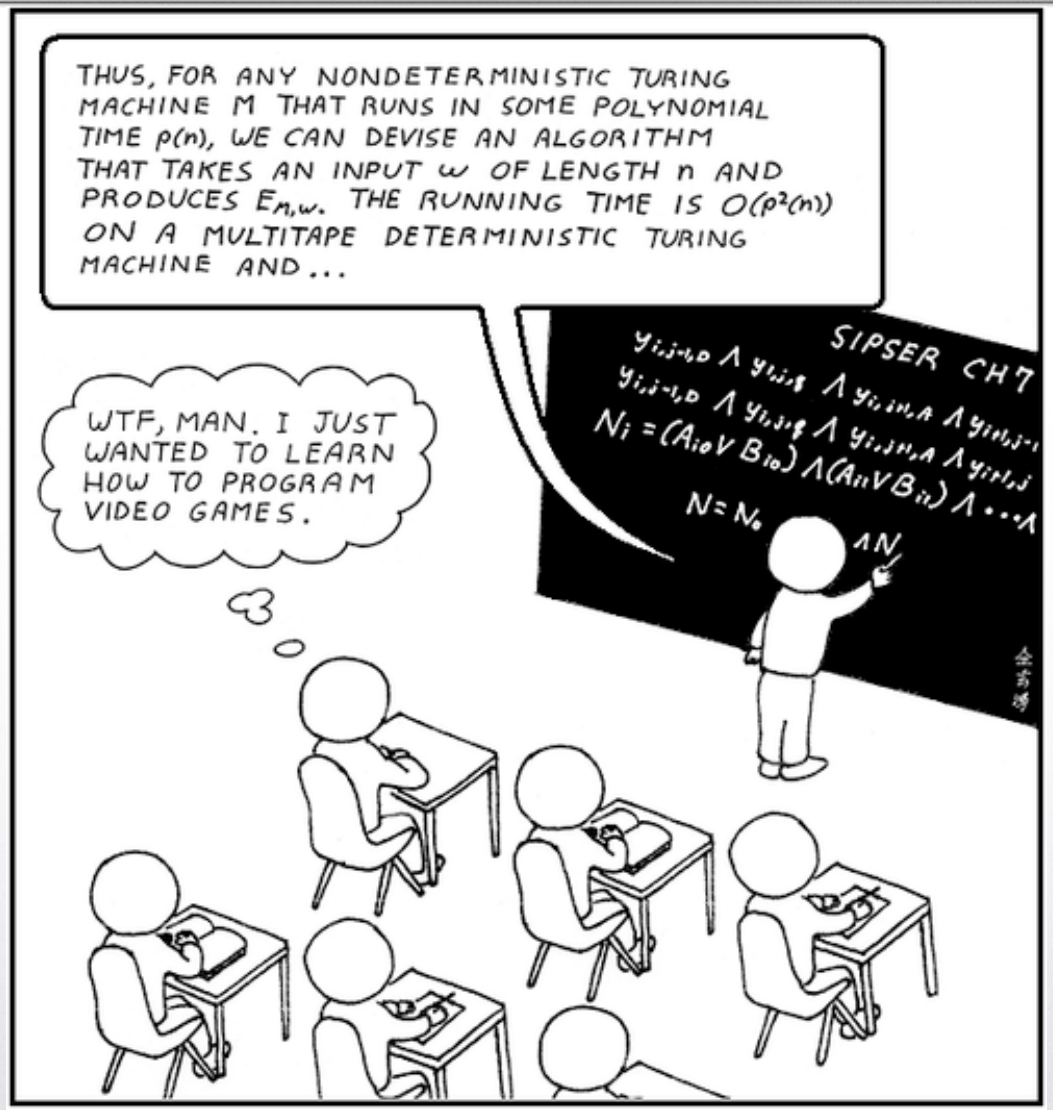
# Il calcolatore moderno

- 1946: è operativo ENIAC (Electronic Numerical Integrator And Computer), progettato da John Presper Eckert and John Mauchly
- 1949: realizzazione di EDVAC
- 1951: UNIVAC I (UNIVersal Automatic Computer I) è il primo computer prodotto a fini commerciali (business e gestionali) negli USA




# Messaggio

- La teoria della calcolabilità
  - snodo di una avventura intellettuale affascinante
  - di interesse per varie discipline
  - determina la nascita dell'informatica moderna


THUS, FOR ANY NONDETERMINISTIC TURING MACHINE M THAT RUNS IN SOME POLYNOMIAL TIME p(n), WE CAN DEVISE AN ALGORITHM THAT TAKES AN INPUT w OF LENGTH n AND PRODUCES $E_{M,w}$. THE RUNNING TIME IS $O(p^2(n))$ ON A MULTITAPE DETERMINISTIC TURING MACHINE AND...
WTF, MAN. I JUST WANTED TO LEARN HOW TO PROGRAM VIDEO GAMES.
SIPSER CH 7
$N = N_0 \wedge N$


# Informazioni sul corso

- Orario
  Lun, Mar (8:30-10:15), primo semestre
- Testo: Nigel Cutland "Computability"
  Cambridge University Press
- Pagina Web:
  http://www.math.unipd.it/~baldan/Computabilita
  + Moodle


# Informazioni sul corso

- Ricevimento: Appuntamento
- Esame
  - Prova intermedia: bonus fino a 3 punti (Nov 18, orario di lezione - da confermare)
  - Scritto (Esercizi) + orale "facoltativo"