

## Informatica

Vittorio Di Tomaso  
www.lett.unipmn.it/~ditomaso/torino  
ditomaso@celi.it

Informatica - 2005

1

## Obiettivi

- Il corso si propone di fornire gli strumenti teorici e metodologici di base per la comprensione dei sistemi di elaborazione dell'informazione non strutturata.

Informatica - 2005

2

## Informazione

- Informazione: è la rappresentazione di "fatti" mediante simboli
  - sintassi: regole per combinare i simboli in espressioni "ben fatte"
  - semantica: relazione tra i simboli ed il referente (agenti, oggetti, proprietà delle azioni)
  - pragmatica: relazione tra il linguaggio e chi lo usa

Informatica - 2005

3

## Informazione

- I dati sono spesso legati ad un supporto
  - carta
  - dischi
  - RAM

Informatica - 2005

4

## Informazione

- L'informazione deve essere fruibile
- Il metodo di gran lunga più utilizzato consiste nella visualizzazione dell'informazione
- Esempio:
  - dati scritti
  - tabelle
  - rappresentazione grafica diretta

Informatica - 2005

5

## Dati scritti

- Rapporto sui dipendenti dell'Azienda X
  - Dal sondaggio svoltosi il giorno 2/10/1996 si è constatato che in tutta l'azienda operano 55 persone delle quali 1 ha compiti direzionali, 2 sono ingegneri con il compito della progettazione dei prodotti, 4 sono impiegati in ufficio per lavori di segreteria, 5 sono addetti al magazzino, 34 sono operai addetti alla costruzione dei prodotti, 3 sono rappresentanti e 6 autisti per il trasporto merci

Informatica - 2005

6

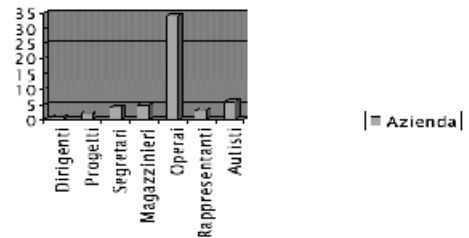
## Dati Tabella

N° impiegati	Attività svolta
1	Attività dirigenziale
2	Ingegneri addetti alla progettazione dei prodotti
4	Impiegati in ufficio per lavori di segreteria
5	Addetti al magazzino
34	Operai addetti alla manutenzione dei prodotti
3	Rappresentanti
6	Autisti
55	Totale

Informatica - 2005

7

## Grafici



Informatica - 2005

8

## Estrazione di informazione

- L'informazione può essere elaborata per produrre nuova informazione
  - La statistica descrittiva cerca di riassumere una grande quantità di dati in poche grandezze rappresentative (media e varianza)
  - L'analisi dei dati permette di scoprire correlazioni e dipendenze inizialmente non note fra variabili o individuare "trend" temporali

Informatica - 2005

9

## Informazione

- L'importanza di manipolare i dati per rendere le informazioni più fruibili per gli esseri umani
- La manipolazione automatica permette di gestire grandi volumi di dati ed espletare funzioni quali:
  - archiviazione
  - visualizzazione
  - trasferimento
  - analisi
  - trasformazione

Informatica - 2005

10

## How much information

- Print, film, magnetic, and optical storage media produced about 5 exabytes of new information in 2002.
- Ninety-two percent of the new information was stored on magnetic media, mostly in hard disks.

[School of Information Management and Systems at the University of California at Berkeley]

Informatica - 2005

11

## How much information

- How big is five exabytes?
- If digitized, the 19 million books and other print collections in the Library of Congress would contain about 10 terabytes of information;
- 5 exabytes of information is equivalent in size to the information contained in half a million new libraries the size of the Library of Congress print collections.

Informatica - 2005

12

## Kilobyte (KB)

- 1,000 bytes OR  $10^3$  bytes
- 2 Kilobytes: A Typewritten page.
- 100 Kilobytes: A low-resolution photograph.

Informatica - 2005

13

## Megabyte (MB)

- 1,000,000 bytes OR  $10^6$  bytes
- 1 Megabyte: A small novel OR a 3.5 inch floppy disk.
- 2 Megabytes: A high-resolution photograph.
- 5 Megabytes: The complete works of Shakespeare.
- 10 Megabytes: A minute of high-fidelity sound.
- 100 Megabytes: 1 meter of shelved books.
- 500 Megabytes: A CD-ROM.

Informatica - 2005

14

## Gigabyte (GB)

- 1,000,000,000 bytes OR  $10^9$  bytes
- 1 Gigabyte: a pickup truck filled with books.
- 20 Gigabytes: A good collection of the works of Beethoven.
- 100 Gigabytes: A library floor of academic journals.

Informatica - 2005

15

## Terabyte (TB)

- 1,000,000,000,000 bytes OR  $10^{12}$  bytes
- 1 Terabyte: 50000 trees made into paper and printed.
- 2 Terabytes: An academic research library.
- 10 Terabytes: The print collections of the U.S. Library of Congress.
- 400 Terabytes: National Climactic Data Center (NOAA) database.

Informatica - 2005

16

## Petabyte (PB)

- 1,000,000,000,000,000 bytes OR  $10^{15}$  bytes
- 1 Petabyte: 3 years of EOS data (2001).
- 2 Petabytes: All U.S. academic research libraries.
- 20 Petabytes: Production of hard-disk drives in 1995.
- 200 Petabytes: All printed material.

Informatica - 2005

17

## Exabyte (EB)

- 1,000,000,000,000,000,000 bytes OR  $10^{18}$  bytes
- 2 Exabytes: Total volume of information generated in 1999.
- 5 Exabytes: All words ever spoken by human beings.

Informatica - 2005

18

## WWW

- 167 TB surface web
- 91.000 TB Deep web
- 440.000 email
- 274 TB IRC
- >530.000 TB totale

## G\*\*gle

- Un esempio di tecnologia informatica che aiuta a utilizzare una grande quantità di informazione non strutturata
- Una tecnologia che sembra semplice, ma che invece è molto complessa
- G\*\*gle non potrebbe esistere senza le tecniche sviluppate in una branca dell'informatica che alcuni definiscono "informatica umanistica"

## Argomenti

- Storia
- Fondamenti teorici
- Algoritmi, dati e programmi
- Information management
- Linguaggi di markup

## Testi

- David Harel, Computer a responsabilità limitata, Einaudi, 2002
- Numerico T., Vespignani A. (a cura di), Informatica per le scienze umanistiche, Il Mulino 2003

## Informatica

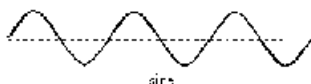
- Lo studio dei calcolatori digitali (digital computers) e dei loro utilizzi
  - Ingegneria (hardware e software)
  - Matematica (algoritmi, complessità)

## Analogico/Digitale

- Due modi per rappresentare un'informazione
- Solitamente un'informazione che deve essere trasmessa
- Continuo / discreto

## Analogo/Digitale

- Sono analogici i mezzi di trasmissione tradizionali (telefono, radio, televisione)
- I contenuti informativi sono rappresentati da una variazione continua in una grandezza (solitamente una corrente elettrica)



Informatica - 2005

25

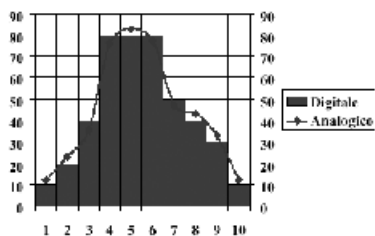
## Analogo/Digitale

- Sono digitali gli strumenti che trattano l'informazione utilizzando un numero finito di stati distinti
- Se sono usati due soli stati, positivo (1) e non positivo (0), si parla di rappresentazione digitale binaria
- I dati sono trasmessi e immagazzinati utilizzando due sole grandezze distinte, ovvero come sequenza di 0 e 1

Informatica - 2005

26

## Analogo/Digitale



Informatica - 2005

27

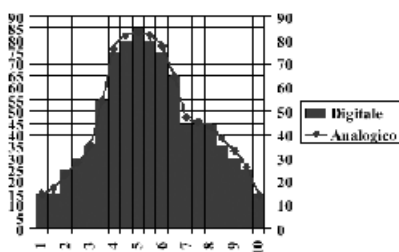
## Analogo/Digitale

- Dato un segnale analogico/continuo se ne può costruire una approssimazione digitale/discreta arbitrariamente vicina

Informatica - 2005

28

## Analogo/Digitale



Informatica - 2005

29

## Informatica

- Lo studio dei calcolatori digitali (digital computers) e dei loro utilizzi
  - Ingegneria (hardware e software)
  - Matematica (algoritmi, complessità)

Informatica - 2005

30

## Fortune-500 2004, Miliardi di US\$

Computers and data services	82,7		
		Eds	21,5
Periferiche	18,7		
		EMC	6,2
Software	59,7		
		Microsoft	32,1
		Oracle	9,4
Equipment	253,2		
		IBM	89,1
		HP	73
		Compaq	
		Dell	41,4
		Sun	11,4
		Xerox	15
Network	71,3		
Semiconduttori	99,6		
		Intel	30,1
Totale	502,5		

Informatica - 2005

31

## Fortune-500 2004, Miliardi di US\$

Telecom	306,5		
Entertainment	116,3		
Petrolio (Raffinerie)	559,2		
		Exxon M.	213,1
Automotive	542		
		GM	195,6
		Ford	164,4
Retail	483,7		
		Wal Mart	258,6

Informatica - 2005

32

## \$\$\$

- Market cap di alcune aziende che manipolano testo non strutturato (miliardi di \$)
  - Google: 60
  - Yahoo: 47
  - Amazon: 13
  - Ebay: 42
- Per confronto:
  - Microsoft: 275 - IBM 123
  - Fiat: 5,9 -

Informatica - 2005

33

## Storia

- L'informatica ha una storia (che precede l'invenzione del termine)
- E' una storia interessante

Informatica - 2005

34

## Storia

- Significato di "computer"
  - < 1920: essere umano che esegue calcoli.
  - 1920 – 1950: calcolatrice elettro-meccanica che esegue alcuni dei calcoli precedentemente eseguiti da esseri umani
  - > 1950: calcolatore digitale programmabile

Informatica - 2005

35

## Babbage

- Professore di Matematica a Cambridge dal 1828 al 1839
- The Difference Engine
  - Macchina calcolatrice in grado di calcolare automaticamente tabelle di numeri (tavole logaritmiche, tavole astronomiche)
  - Interamente meccanica
  - Mai completata

Informatica - 2005

36

## Babbage

- The Analytical Engine
  - Macchina calcolatrice universale
  - Dotata di memoria centrale e di un'unità di elaborazione separata
  - Controllata da un programma immagazzinato su schede perforate
  - Mai costruita, ma fonte di grande interesse teorico e tecnologico

Informatica - 2005

37

## Analog computers

- Vari esempi di macchine in grado di calcolare integrali, usate anche durante la I guerra mondiale per calcoli di traiettorie
- Analogici in quanto rappresentano le informazioni attraverso differenze (continue) di potenziale elettrico

Informatica - 2005

38

## Universal Turing Machine

- 1935, Cambridge: Alan Turing descrive il moderno computer digitale
- Una macchina astratta consistente in una memoria (infinita) e un meccanismo di lettura/scrittura capace di leggere e scrivere simboli sulla memoria
- Le azioni di lettura / scrittura sono descritte da un programma a sua volta contenuto nella memoria

Informatica - 2005

39

## Calcolatori elettromeccanici

- Basati su rele': interruttori meccanici controllati elettricamente
- Contengono parti meccaniche: lenti e inaffidabili
- Molti esemplari durante la II Guerra Mondiale

Informatica - 2005

40

## Calcolatori elettronici

- Basati originariamente su valvole (tubi a vuoto) in cui si muovono elettroni
- Nessuna parte mobile
- Velocità di svolti ordini di grandezza superiori
- Thomas Flowers, British Post Office, 1934

Informatica - 2005

41

## Atanasoff

- Iowa State University 1937-1942
- Costruzione di calcolatori general purpose basati su tubi a vuoto
- ABC (Atanasoff-Berry Computer) 1939: 300 tubi a vuoto
- Mai funzionante a causa di insufficienti sistemi di lettura di schede perforate
- Progetto terminato nel 1942

Informatica - 2005

42

## Critto-analisi

- La seconda guerra mondiale da' una spinta enorme alla ricerca
- Bletchey Park e il gruppo di critto-analisti (compreso Turing)

## Colossus

- Sviluppato per la decodifica di Fish
- Elettronico: 1600-2400 tubi a vuoto
- Nessun programma in memoria: per riprogrammarlo vengono modificate le configurazioni dei circuiti
- Non general purpose: solo funzioni numeriche per critto-analisi
- Abbrevio' la guerra di due anni (!)

## Dopo la guerra: UK

- ACE (Automatic Computing Machine), il primo design completo di Alan Turing
- The Manchester Baby, programmabile!

## Dopo la guerra: USA

- ENIAC, simile a Colossus, special purpose per calcoli balistici, 1945
- John Von Neuman: EDVAC
  - General purpose
  - Stored program

## Fino ad oggi

- Da questo punto in avanti è stata in gran parte questione di:
  - Avanzamenti tecnologici (in primo luogo memorie ad alta velocità)
  - Esigenze applicative
  - Interazioni tra i due precedenti

## Sviluppo del computer digitale

Nome	Modello	Modello di	Caratteristiche
1943	Colossus	Special Purpose	Primo computer elettronico per critto-analisi
1945	ENIAC	Special Purpose	Primo computer elettronico per calcoli balistici
1946	EDVAC	General Purpose	Primo computer elettronico per calcoli generali
1947	Manchester Baby	General Purpose	Primo computer elettronico programmabile
1948	Manchester Mark I	General Purpose	Primo computer elettronico programmabile con memoria
1949	UNIVAC	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1951	ENIAC II	Special Purpose	Primo computer elettronico per calcoli balistici
1952	UNIVAC II	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1953	IBM 701	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1954	IBM 702	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1955	IBM 703	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1956	IBM 704	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1957	IBM 705	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1958	IBM 706	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1959	IBM 707	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1960	IBM 708	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1961	IBM 709	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1962	IBM 7090	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1963	IBM 7090A	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1964	IBM 7090B	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1965	IBM 7090C	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1966	IBM 7090D	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1967	IBM 7090E	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1968	IBM 7090F	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1969	IBM 7090G	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1970	IBM 7090H	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1971	IBM 7090I	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1972	IBM 7090J	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1973	IBM 7090K	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1974	IBM 7090L	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1975	IBM 7090M	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1976	IBM 7090N	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1977	IBM 7090O	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1978	IBM 7090P	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1979	IBM 7090Q	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1980	IBM 7090R	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1981	IBM 7090S	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1982	IBM 7090T	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1983	IBM 7090U	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1984	IBM 7090V	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1985	IBM 7090W	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1986	IBM 7090X	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1987	IBM 7090Y	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1988	IBM 7090Z	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1989	IBM 7090AA	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1990	IBM 7090AB	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1991	IBM 7090AC	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1992	IBM 7090AD	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1993	IBM 7090AE	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1994	IBM 7090AF	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1995	IBM 7090AG	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1996	IBM 7090AH	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1997	IBM 7090AI	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1998	IBM 7090AJ	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
1999	IBM 7090AK	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2000	IBM 7090AL	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2001	IBM 7090AM	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2002	IBM 7090AN	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2003	IBM 7090AO	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2004	IBM 7090AP	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2005	IBM 7090AQ	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2006	IBM 7090AR	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2007	IBM 7090AS	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2008	IBM 7090AT	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2009	IBM 7090AU	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2010	IBM 7090AV	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2011	IBM 7090AW	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2012	IBM 7090AX	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2013	IBM 7090AY	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2014	IBM 7090AZ	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2015	IBM 7090BA	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2016	IBM 7090BB	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2017	IBM 7090BC	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2018	IBM 7090BD	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2019	IBM 7090BE	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2020	IBM 7090BF	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2021	IBM 7090BG	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2022	IBM 7090BH	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2023	IBM 7090BI	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2024	IBM 7090BJ	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo
2025	IBM 7090BK	General Purpose	Primo computer elettronico programmabile con memoria e sistema operativo



## Informatica

- Architetture hardware
- Architetture software
- Architetture di sistemi (Integrazione hardware e software)

Informatica - 2005

49

## Informatica

- Software: i programmi o gli insiemi di istruzioni che dicono all'hardware che cosa deve fare
  - Ingegneria del software
  - Linguaggi di programmazione
  - Sistemi operativi
  - Database
  - Intelligenza artificiale
  - Grafica
  - Applicazioni

Informatica - 2005

50

## Informatica

- Teoria e matematica
  - Metodi computazionali
  - Analisi numerica
  - Strutture dati
  - Algoritmi

Informatica - 2005

51

## Informatica

- Basi teoriche
  - Numeri binari (codifica binaria dell'informazione)
  - Algebra booleana (operazioni su numeri binari)
  - Alan Turing (la macchina universale)
  - John Von Neuman (memoria di programma)

Informatica - 2005

52

## Informatica

- Avanzamenti tecnologici
  - Transistor (1947, Bell Lab)
  - Miniaturizzazione (da migliaia a milioni di transistor su un chip)
  - Immagazzinamento magnetico / ottico (dai megabyte ai terabyte)
  - Reti (circa 2,5 miliardi di indirizzi IP)

Informatica - 2005

53

## Informatica

- Esigenze degli sviluppatori
  - Linguaggi assembler
  - Sistemi operativi
  - Linguaggi di alto livello

Informatica - 2005

54

## Informatica

- Esigenze degli utilizzatori
  - Calcoli scientifici
  - Applicazioni business
  - Applicazioni consumer
  - Telecomunicazioni
  - Convergenza sui media digitali (?)

Informatica - 2005

55

## Informatica

- I sistemi informatici sono sempre più complessi, più difficile da capire e da tenere sotto controllo
- Lo sviluppo dei sistemi informatici richiede metodologie rigorose

Informatica - 2005

56

## A che cosa serve

- Un computer trasforma dati (rappresentazioni di fatti in forma numerica) in informazioni utili
- Con tali informazioni è possibile:
  - Risolvere problemi
  - Aumentare la produttività
  - Risparmiare tempo
- Il software consente la trasformazione

Informatica - 2005

57

## Che cosa è

- Un computer esegue due tipi di operazioni:
  - Aritmetiche (calcoli sui numeri, addizioni, sottrazioni...)
  - Logiche (confronti tra numeri)
- Ogni processo computazionale è costruito attorno a questi due tipi di operazioni

Informatica - 2005

58

## Che cosa è

- Il software contiene le regole che sono date al computer per eseguire operazioni aritmetiche e logiche
- Il software dirige le computazioni per trasformare dati in informazioni utili

Informatica - 2005

59

## Software

- Qualunque tipo di dato sia possibile inserire in un moderno computer (numeri, testi, audio, video) deve essere trasformato in forma numerica binaria
- I circuiti del computer comprendono due sole informazioni: 1 e 0

Informatica - 2005

60

## Software

- Qualsiasi dato rappresentato internamente ad un computer è rappresentato in forma binaria
- Allo stesso modo qualsiasi programma (insieme di istruzioni da eseguire sui dati) è rappresentato in forma binaria

Informatica - 2005

61

## Software

- La scrittura di dati e programmi direttamente in forma binaria è poco adatto agli esseri umani
- Lo sviluppo di maniere più adatte alle persone per interagire con i computer è stato ed è un forza dominante nello sviluppo dell'informatica

Informatica - 2005

62

## Software di sistema

- Il software di sistema controlla le attività interne del computer
  - Linguaggi di programmazione
  - Sistemi operativi
  - Utilità

Informatica - 2005

63

## Linguaggi di programmazione

- I linguaggi di programmazione sono utilizzati per descrivere insiemi di istruzioni per il computer
- I linguaggi sono molti e molto diversi, ma alla fine qualunque programma scritto in qualunque linguaggio deve essere convertito in forma binaria
- Linguaggi interpretati vs linguaggi compilati

Informatica - 2005

64

## Linguaggi

- La prima generazione: linguaggi macchina
- Programmi direttamente codificati in codice binario

Informatica - 2005

65

## Linguaggi

- La seconda generazione: linguaggi assembler
- A partire dagli anni 50
- Utilizzo di simboli invece che di numeri binari
- Totalmente legato all'hardware

Informatica - 2005

66

## Linguaggi

- La terza generazione: linguaggi di alto livello
- Parzialmente indipendenti dalla piattaforma hardware, quindi riutilizzabili
- Solitamente compilati, per mezzo di specifici programmi, in linguaggio assembler

Informatica - 2005

67

## Linguaggi

- La quarta generazione: linguaggi di alto livello interpretati e compilati

Informatica - 2005

68

## Linguaggi

- Lo sviluppo dei linguaggi di alto livello ha consentito lo sviluppo dell'industria del software come la conosciamo
- IBM 1970: il software è venduto separatamente dall'hardware

Informatica - 2005

69

## Sistemi operativi

- Il sistema operativo comprende un insieme di programmi di sistema che gestiscono tutte le operazioni di sufficientemente basso livello, quali l'interazione con le periferiche di input e di output
- Il sistema operativo è caricato all'avvio del computer
- Tra le altre cose, esegue altri programmi e ne gestisce le interazioni

Informatica - 2005

70

## Programmi di utilità

- Espandono le capacità del sistema operativo
- La differenza non è sempre chiara:
  - il gestore delle finestre è parte del kernel di Windows
  - Il gestore delle finestre non è parte del kernel di Linux

Informatica - 2005

71

## Software applicativo

- Eseguono funzioni specializzate per specifici obiettivi
- La variabilità è infinita: (quasi) qualunque compito oggi ha un software di supporto
- Tale ubiquità del software è un bene o è un male?

Informatica - 2005

72

## Applicazioni

- Fogli di calcolo
- Database management
- Word processing
- Reti
- Intrattenimento

Informatica - 2005

73

## Che cosa è un computer

- Un computer è una macchina che risolve problemi eseguendo istruzioni
- Una sequenza di istruzioni che descrivono come eseguire un certo compito è un programma

Informatica - 2005

74

## Che cosa si può fare

- Non è vero che:
  - Un computer, se mettete il programma appropriato, farà *tutto* quello che desiderate
- Quali sono le cose che si possono fare?
- Quali sono le cose che non si possono fare?

Informatica - 2005

75

## Che cosa si può fare

- Algoritmi
- Programmi
- Processi
  
- Input
- Output

Informatica - 2005

76

## Che cosa si può fare

- Calcola la media dei voti degli studenti studente
- Input: un elenco di file contenenti la media di ciascuno studente
- Output: un numero

Informatica - 2005

77

## Algoritmi

1. Scrivi 0 nella casella totale
2. Leggi i file dalla lista aggiungendo la media al totale
3. Raggiunto l'ultimo dei file produci come output il risultato della divisione del totale per il numero dei file letti

Informatica - 2005

78

## Programmi

- Realizzate l'algoritmo nel vostro linguaggio di programmazione preferito

Informatica - 2005

79

## Processi

Studiante	Media	Totale	
File1	30	0	
File2	27	57	
.			
.			
FileN	21	Tot	
			Tot/N

Informatica - 2005

80

## Input

- L'algoritmo funziona per input arbitrari (università grandi, piccole o infinite)
- Il tempo di elaborazione del programma può tuttavia variare
- Gli input devono essere *ammissibili*

Informatica - 2005

81

## Problemi algoritmici

- Per specificare un problema algoritmico dobbiamo specificare:
  - L'insieme degli input ammissibili
  - L'insieme degli output desiderati in funzione degli input

Informatica - 2005

82

## Problema 1

- Input: due numeri interi J e K
- Output: il numero  $J^2+3K$

Informatica - 2005

83

## Problema 2

- Input: Un intero positivo K
- Output: la somma di tutti gli interi da 1 a K

Informatica - 2005

84

### Problema 3

- Input: un intero positivo K
- Output: “SI” se K è primo, “NO” altrimenti

Informatica - 2005

85

### Problema 4

- Input: una lista di parole L in una lingua fissata
- Output: la lista L in ordine alfabetico (lessicografico)

Informatica - 2005

86

### Problema 5

- Input: due testi in una lingua fissata
- Output: un lista di parole comuni ai due testi

Informatica - 2005

87

### Problema 6

- Input: una carta stradale con un certo numero di città collegate tra loro da strade, la cui lunghezza è indicata sulla carta stessa, e una coppia di città fissate A e B
- Output: una descrizione del tragitto più breve che unisce A con B

Informatica - 2005

88

### Problema 7

- Input: una carta stradale con un certo numero di città collegate tra loro da strade, la cui lunghezza è indicata sulla carta stessa, e un numero fissato K
- Output: “SI” se esiste un itinerario che tocca tutte le città la cui lunghezza non supera K, no altrimenti

Informatica - 2005

89

### Problema 8

- Input: un programma P scritto in Java, due variabili intere X (input di P) e Y (output di P) e un numero K
- Output: il numero 2K se P pone sempre  $Y=X^2$ , il numero 3K se ciò non accade

Informatica - 2005

90

## Tipi di problemi

- Strettamente computazionali (1,2)
- Riordinamento (4)
- Riconoscimento (5)
- Ottimizzazione (6)
- Decisione (3,7)
- Ibridi (8)

Informatica - 2005

91

## Problemi algoritmici

- Input infiniti (l'algoritmo deve andare bene per tutti gli input ammissibili)
- Spesso è difficile definire l'output (come definire la "mossa migliore" data una configurazione della scacchiera?)
- Talvolta è difficile definire l'input (come far distribuire 200.000 giornali a 1.000 rivendite in 100 città usando 50 furgoni?)
- Talvolta è difficile definire sia input che output

Informatica - 2005

92

## Problemi algoritmici

- Un problema è risolto se si riesce a trovare un algoritmo appropriato
- Ovvero un algoritmo che fornisce risultati corretti per tutti gli input ammissibili

Informatica - 2005

93

## Algoritmi e programmi

- Un programma è una formulazione precisa dell'algoritmo che può essere eseguita dal computer
- Un programma è scritto in un linguaggio di programmazione
- Ne parleremo più avanti...

Informatica - 2005

94

## Algoritmi e programmi

- Un algoritmo è corretto quando fornisce l'output giusto per tutti gli input ammissibili
- Le soluzioni parziali non sono accettabili
- Le soluzioni sbagliate non sono accettabili
- Soluzioni parziali e/o sbagliate possono avere conseguenze catastrofiche

Informatica - 2005

95

## Algoritmi e programmi

- Oltre che dare output corretti, chiediamo che l'esecuzione di un programma a partire da un input ammissibile termini in un intervallo di tempo finito
- E' possibile che in certi casi il programma non termini

Informatica - 2005

96



## Problemi algoritmici

- Problemi risolvibili
- Problemi non risolvibili
- Problemi risolvibili, ma è troppo costoso farlo
- Problemi che non sappiamo se siano risolvibili

Informatica - 2005

97

## A volte si può fare

- Sono risolvibili tutti i problemi risolti
- Esistono algoritmi che danno l'output corretto per tutti gli input ammissibili in un tempo finito

Informatica - 2005

98

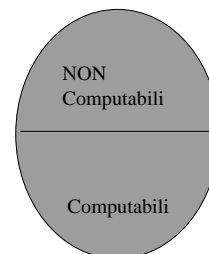
## A volte è impossibile

- Esistono problemi (algoritmici) che nessuna macchina esistente o futura sia in grado di risolvere, qualunque programmi utilizzi e anche avendo a disposizione risorse infinite
- Per certi problemi algoritmici non esistono soluzioni

Informatica - 2005

99

## A volte è impossibile



Informatica - 2005

100

## I problemi finiti sono risolvibili

- I problemi con un insieme finito di input ammissibili sono risolvibili
- Basta scrivere una tabella che ad ogni input faccia corrispondere l'output corretto
- L'intero problema può essere inserito nell'algoritmo (inserendo le coppie input → output, che sono in numero finito)
- NB: non è detto che sia "facile"

Informatica - 2005

101

## Problemi interessanti

- I problemi "interessanti" sono quelli con infiniti input ammissibili
- Per i problemi con infiniti input ammissibili non sappiamo a priori se esista un algoritmo *finito* che risolva tutti gli *infiniti* casi possibili

Informatica - 2005

102

## Il problema del ricoprimento

- Un problema non computabile riguarda la copertura di una superficie con vari tipi di piastrelle colorate
- Ogni piastrella è un quadrato di uguale dimensione (unitaria) diviso in quattro dalle diagonali
- Ciascuna delle quattro parti è colorata da un colore diverso
- L'orientamento è fissato e non possono essere ruotate

Informatica - 2005

103

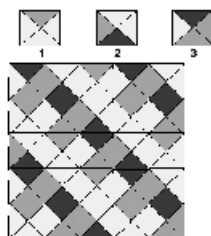
## Il problema del ricoprimento

- Input: Un insieme finito T di tipi di piastrelle, ciascuno definito dai suoi quattro colori; una porzione finita di piano
- Output: "SI" se è possibile coprire la porzione di piano scelta con le piastrelle T soddisfacendo la restrizione che i lati di due piastrelle adiacenti devono avere lo stesso colore, "NO" altrimenti. È possibile usare infinite piastrelle.

Informatica - 2005

104

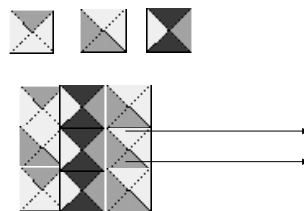
## Ricoprimento: la risposta è SI



Informatica - 2005

105

## Ricoprimento: la risposta è NO



Informatica - 2005

106

## Ricoprimento

- Nel primo caso, con l'insieme T si può ricoprire qualunque porzione finita del piano, grande a piacere
- Nel secondo caso, già una porzione di lato di lunghezza 3 non può essere ricoperta
- Esiste un algoritmo che distingue sempre i due casi?

Informatica - 2005

107

## Il ricoprimento non è decidibile

- Non esiste e non esisterà mai un algoritmo che risolva il problema del ricoprimento
- Il problema del ricoprimento è **non computabile**
- Poiché cerchiamo un algoritmo di decisione, diciamo che è **indecidibile**

Informatica - 2005

108

## Attenzione!

- Questa non è una soluzione:
  - Scrivi SI se le piastrelle T sono in grado di ricoprire l'area
  - Scrivi NO altrimenti
- Non lo è perché non esiste alcun modello di computazione in cui lo sia

Informatica - 2005

109

## Modelli computazionali

- La macchina di Turing è un modello computazionale molto semplice
- Tuttavia, una macchina di Turing è in grado di risolvere tutti i problemi algoritmici effettivamente risolvibili

Informatica - 2005

110

## Tesi di Church-Turing

- L'insieme dei problemi risolvibili da una macchina di Turing e l'insieme dei problemi effettivamente risolvibili coincidono
- Tutti i computer e i linguaggi, se forniti di tempo e spazio illimitato, sono equivalenti.

Informatica - 2005

111

## Tesi di Church-Turing

- E' una tesi, non si dimostra
- Tuttavia è solida:
  - Ci sono molte definizioni di computabile, e sono tutte equivalenti (definiscono lo stesso insieme come insieme dei problemi risolvibili effettivamente)
- Possiamo fidarci della distinzione tra computabile e non computabile

Informatica - 2005

112

## Il problema della fermata

- Non è possibile dire (a priori) se un programma termina (in generale o su un certo input)

Informatica - 2005

113

## Il problema della fermata

- Per alcuni programmi si può sapere:
  1. Finché  $X \neq 1$ , poni  $X = X - 2$
  2. Fermati
- Se  $X$  è un numero intero, si arresta soltanto se  $X$  è dispari. Se  $X$  è pari va avanti all'infinito

Informatica - 2005

114

## Il problema della fermata

- Finché  $X \neq 1$  esegui:
  1. Se  $X$  è pari allora  $X = X/2$
  2. Se  $X$  è dispari allora poni  $X = 3X + 1$
- Il programma termina se il numero è dispari
- Se il numero è pari... o termina, va avanti fornendo output sempre diversi
- Non siamo certi che non termini (gli output sono sempre diversi), ma non sembra terminare...

Informatica - 2005

115

## Il problema della fermata

- Dato un programma  $A$  e un input potenziale  $X$ , ci chiediamo se  $A$  termini dato  $X$
- Il problema è indecidibile
- Non c'è modo di sapere, nel caso generale e in tempo finito, se l'esecuzione di un dato programma avrà termine partendo da un certo input

Informatica - 2005

116

## Problemi indecidibili

- Nessuna proprietà non banale dei programmi è decidibile
- Una proprietà fondamentale, la correttezza del programma, è *fortemente* indecidibile

Informatica - 2005

117

## A volte è troppo costoso...

- Esistono problemi che sappiamo essere computabile e per i quali conosciamo anche gli algoritmi risolutori
- Tuttavia tali problemi di fatto non possono essere risolti in pratica
- La loro risoluzione richiede risorse che non sono disponibili

Informatica - 2005

118

## Risorse e complessità

- Le risorse che vengono consumate dai programmi sono
  - Il tempo di esecuzione
  - La quantità di memoria necessaria
- Tempo (di esecuzione) e spazio (di memoria) sono le misure della **complessità computazionale**

Informatica - 2005

119

## Risorse e complessità

- Il tempo si misura calcolando il numero di "azioni elementari" che un programma esegue per ottenere il risultato
- Lo spazio si misura calcolando l'area di memoria in cui deve immagazzinare i dati
- Le misure di tempo e spazio dipendono dalla macchina in uso

Informatica - 2005

120

## Risorse e complessità

- Tempo e spazio sono solitamente variabili in funzione dell'input
- L'algoritmo che calcola le medie esegue ad ogni passo due operazioni (leggi il file, somma al totale)
- Poiché i passi sono  $N$ , la sua complessità sarà  $2N$

Informatica - 2005

121

## Risorse e complessità

- Studiare la complessità di un programma significa chiedersi come varia l'occupazione di risorse al variare dell'input
- Lo spazio si può (entro certi limiti) aumentare
- Il tempo in molti casi è cruciale (in un ABS può, letteralmente, fare la differenza tra evitare e colpire l'ostacolo)

Informatica - 2005

122

## Migliorare l'efficienza

- Per verificare se un nome è in una lista (non ordinata), ad ogni passo:
  - Leggi un nome e verifica se è quello cercato
  - Se lo è, allora rispondi SI
  - Se non lo è, controlla se sei a fine lista. Se sei a fine lista allora rispondi NO
- Il tempo sarà  $2N$ , immaginando che la verifica di uguaglianza e la verifica di fine lista siano due operazioni elementari

Informatica - 2005

123

## Migliorare l'efficienza

- Per verificare se un nome è in una lista (non ordinata), aggiungi il nome alla lista e, ad ogni passo:
  - Leggi un nome e verifica se è quello cercato
  - Se lo è, controlla se sei a fine lista. Se sei a fine lista allora rispondi NO, altrimenti rispondi SI
- Il tempo sarà  $N$ , la verifica di fine lista viene fatta una volta sola

Informatica - 2005

124

## Il caso peggiore

- Spesso la cosa più interessante è calcolare il caso peggiore
- L'algoritmo di ricerca (prima versione) esegue  $2N$  operazioni solo se l'elemento cercato è a fine lista
- In tutti gli altri casi ne esegue di meno
- Trovato il caso peggiore, saremo sicuri che il nostro programma non potrà fare peggio di così, qualunque sia l'input

Informatica - 2005

125

## Complessità lineare

- Quando il tempo cresce proporzionalmente rispetto all'input, si dice che il programma gira in **tempo lineare**
- La linearità si denota con l'espressione  $O(n)$  e significa che il tempo di esecuzione è limitato superiormente da  $KN$ , dove  $K$  è una costante positiva
- Ci sono molti casi in cui il tempo è inferiore a  $KN$ , ma nessuno in cui è superiore

Informatica - 2005

126

## La ricerca binaria

- Input: un lista ordinata di parole L e una parola p
- Output: "SI" se  $p \in L$ , "NO" altrimenti
- Prendi la parola  $p_i$  che si trova a metà di L a metà
  - Se  $p = p_i$  rispondi SI e termina
  - Se  $p > p_i$  ripeti il procedimento con la parte di L che contiene le parole maggiori di  $p_i$
  - Se  $p < p_i$  ripeti il procedimento con la parte di L che contiene le parole minori di  $p_i$

Informatica - 2005

127

## Complessità logaritmica

- Consideriamo, ai fini della complessità, come operazione significativa l'operazione di confronto tra p e  $p_i$
- Poiché ad ogni passo dividiamo l'input a metà, nel caso peggiore la complessità è proporzionale a  $\log_2 N$

Informatica - 2005

128

## Complessità logaritmica

N	Logaritmica	Lineare
10	4	10
100	7	100
1000	10	1000
$10^6$	20	$10^6$
$10^9$	30	$10^9$
$10^{18}$	60	$10^{18}$

Informatica - 2005

129

## Ordinamento

- Input: una lista L di parole
- Output: la lista  $L_1$  ordinata alfabeticamente
- Ad ogni passo
  - Cerca l'elemento minimo di L, rimuovilo e ponilo in  $L_1$
  - Se L è vuota fermati, altrimenti ripeti il passo precedente

Informatica - 2005

130

## Complessità quadratica

- L'operazione di ricerca del minimo costa N, perché nel caso peggiore richiede di controllare tutti gli elementi della lista
- Tale operazione di ricerca viene ripetuta su un input sempre più piccolo:
  - $(N-1) + (N-2) + \dots + 0$
- La complessità è  $N^2/2$ ,  $O(N^2)$ , ovvero quadratica

Informatica - 2005

131

## Complessità dell'ordinamento

- NB: esistono algoritmi di ordinamento migliori:  $N \log_2 N$
- Il che vuol dire che per ordinare un milione di elementi servono 20 milioni di mosse invece che 500 milioni.

Informatica - 2005

132

## Limiti superiori e inferiori

- Ogni problema ha una “complessità intrinseca”, a noi ignota
- Dato un problema, non appena troviamo un algoritmo che lo risolve e ne calcoliamo la complessità, abbiamo trovato un limite superiore
- Possiamo sempre cercare soluzioni migliori, a meno che non siamo in grado di dimostrare che, considerando tutti gli algoritmi possibili, non si può fare di meglio
- In questo caso abbiamo trovato un limite inferiore

Informatica - 2005

133

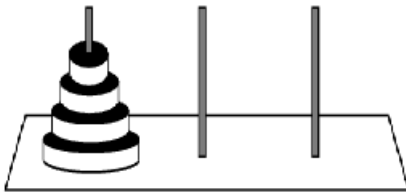
## Soluzioni ottimali

- Per alcuni problemi (che diciamo chiusi) conosciamo il limite inferiore
- Un algoritmo che risolve quel problema con complessità pari al limite inferiore è ottimale (a meno di una costante)
- Meglio così non si può fare
- Molti problemi sono aperti, nel senso che i limiti inferiori e superiori a noi noti sono diversi

Informatica - 2005

134

## Torre di Hanoi



Informatica - 2005

135

## Torre di Hanoi

- Spostare tutti i pioli da A a B rispettando le regole:
  - Si sposta un disco alla volta
  - Non si può mettere un disco grande sopra uno piccolo
  - Si può usare C come deposito provvisorio

Informatica - 2005

136

## Torre di Hanoi

- Tre pioli, tre dischi, soluzione in 7 mosse:
  - A → B
  - A → C
  - B → C
  - A → B
  - C → A
  - C → B
  - A → B
- 4 dischi, 15 mosse

Informatica - 2005

137

## Torre di Hanoi

- Esiste un algoritmo generale?
- Input: un numero N di dischi
- Output: la sequenza di mosse che risolvono il problema

Informatica - 2005

138

## Torre di Hanoi

- 1. Ripeti fino a quando 1.2 non è più eseguibile:
  - 1.1 Sposta il disco più piccolo sul piolo più vicino in senso orario
  - 1.2 Fai l'unica mossa possibile che non sposti il disco più piccolo
- Fermati

Informatica - 2005

139

## Torre di Hanoi

- Il numero delle mosse è  $2^N - 1$
- Il consumo di tempo cresce esponenzialmente rispetto all'input:  $O(2^N)$
- E' anche un limite inferiore, quindi la soluzione è ottimale
- E' anche una soluzione praticabile?

Informatica - 2005

140

## Torre di Hanoi

- La versione originale prevede 64 dischi, spostati da una squadra di monaci tibetani
- Impiegando un secondo a mossa ci vogliono 500.000 anni per finire
- Impiegando 5 secondi a mossa ci vogliono quasi 3.000 miliardi di anni per finire
- Per fortuna, perché finita l'impresa il mondo finirà

Informatica - 2005

141

## Confronto tra ordini di grandezza

Informatica - 2005

142

## Confronto

- Confronto tra durate con input piccoli ipotizzando 1.000.000 di passi al secondo
- Se si migliora una polinomiale di un fattore 10, si ottiene quasi sempre un risultato accettabile
- Non c'è maniera di rendere accettabile una esponenziale (neanche migliorando di un fattore 10.000)

Informatica - 2005

143

## Confronto tra ordini di grandezza

	10	20	50	100	200
$N^2$	1/10.000 sec	1/2.500 sec	1/400 sec	1/100 sec	1/25 sec
$N^5$	1/10 sec	3,2 sec	5,2 min	2,8 ore	3,7 giorni
$2^N$	1/1000 sec	1 sec	35,7 anni	> 4x114 secoli	Un numero di secoli di 45 cifre
$N^8$	2,8 ore	3,3 x112 anni	Un numero di secoli di 70 cifre	Un numero di secoli di 185 cifre	Un numero di secoli di 445 cifre

Informatica - 2005

144



## Intrattabilità

- Un problema per cui esiste una soluzione “buona”, ovvero polinomiale (lineare, quadratica, logaritmica e simili), è trattabile
- Gli altri sono intrattabili

Informatica - 2005

145

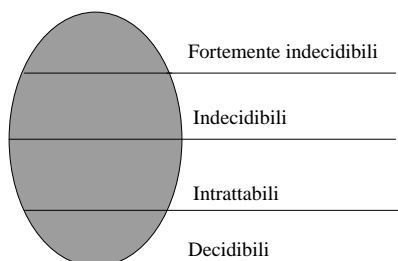
## Intrattabilità

- La distinzione tra problemi trattabili e problemi intrattabili è ben definita
- Si dimostra che i modelli computazionali sono correlati polinomialmente
  - Se un problema è risolvibile in un modello, lo sarà anche in qualsiasi altro e la differenza di tempo sarà descritta da una funzione polinomiale

Informatica - 2005

146

## Problemi algoritmici (II)



Informatica - 2005

147

## Il gioco del tris

- Il primo giocatore ha 9 mosse possibili
- Il secondo 8
- Poi 7 e così via
- Il numero complessivo di mosse è  $9*(9-1)*(9-2)\dots$  ovvero  $9!$  Ovvero 362.880
- Si possono esaminare tutte

Informatica - 2005

148

## Il gioco degli scacchi

- Bianco al primo turno: 20 mosse
- Nei turni successivi sono circa 35 per turno
- In una partita si giocano anche 80 o 100 turni
- Le mosse possibili sono in totale circa  $35^{100}$
- Non si può fare: il computer può vincere ma non può avere una strategia vincente che lo garantisca sempre

Informatica - 2005

149

## Problemi NP-completi

- Problemi in cui si devono esaminare tutte le possibilità e tornare indietro quando si sbaglia
  - Ricerca del cammino più breve
  - Matching
  - Ottimizzazioni
- Problemi con soluzioni polinomiali non deterministiche e soluzioni deterministiche esponenziali

Informatica - 2005

150