

Fondamenti di Informatica

Luca Mottola

Slide credits: Dino Mandrioli, Luciano Baresi,
and many others...

Obiettivi e organizzazione del corso

- Fornire un'introduzione all'informatica con enfasi sulle basi concettuali
- Seguiranno corsi di approfondimento

Traccia del programma

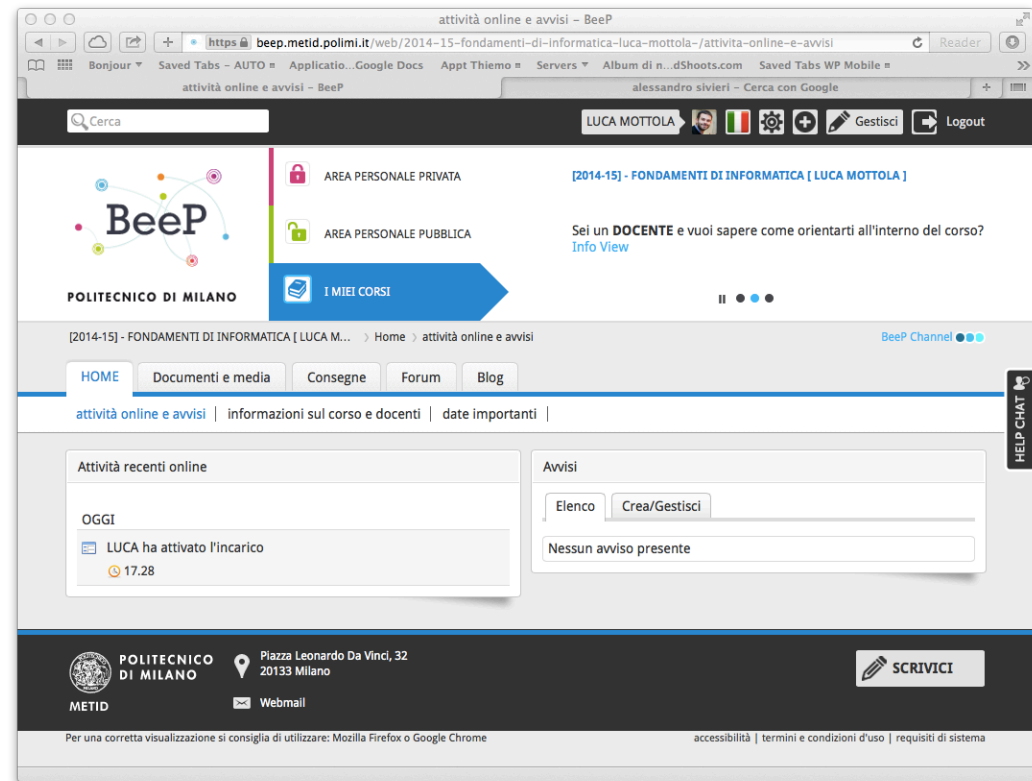
- Concetti introduttivi
- Rappresentazione dell'informazione
- Breve panoramica dei sistemi e applicazioni dell'informatica
- Elementi essenziali della programmazione
 - ...in linguaggio C
- Alcuni aspetti avanzati
 - ...peraltro, i più divertenti

Aspetti organizzativi

- **Lezioni** (54 ore)
- **Esercitazioni** (40 ore)
- **Laboratorio** (20 ore)
- Non seguiremo una calendario regolare di lezioni ed esercitazioni
- I laboratori avverranno solo in date specifiche

Piattaforma Beep

- **beep.metid.polimi.it**
 - avvisi generali, calendario
 - lucidi, temi d'esame, ...
 - forum e blog



Materiale didattico

- **Libro di testo**

- Ceri, Mandrioli, Sbattella: “Informatica: Programmazione”, II edizione, Mc-Graw-Hill Italia

- **Parti di**

- Pelagatti: “Sistema Operativo LINUX”, III° Edizione Agosto 2009, Esculapio editore

Valutazione

- Il **laboratorio** ammette all'esame
 - dà un punteggio (se positivo) da 1 a 3
 - il punteggio è basato su partecipazione e svolgimento di esercizi in aula, durante tutto l'anno
 - chi non prendesse almeno 1 dovrà rifrequentare il corso
 - il voto di laboratorio ha validità di un anno accademico
- Due **prove scritte** intermedie danno fino a 15 punti
 - 9 è il minimo per la sufficienza nelle due prove
 - la sufficienza nella prima prova ammette alla seconda
- Gli **appelli regolari** danno fino a 30 punti
- In corrispondenza del primo appello verrà proposto a tutti un voto
 - basato sulla somma delle due prove intermedie o sul primo appello
 - il voto proposto potrà essere rifiutato
- **Lode** conferita se e solo se il voto finale è ≥ 32

Team

- Luca Mottola
- `home.dei.polimi.it/mottola`
- `luca.mottola@polimi.it`
 - aggiungere [FONDAMENTI] al subject se non volete finire nello spam
- Ufficio @ edificio 22, terzo piano, stanza 319
- Tel 02 2399 3583
- Ricevimento su appuntamento
- Esercitazioni: Alessandro Sivieri
- Lab: da definire



Iniziamo?

Informatica

- Scienza della rappresentazione ed elaborazione rigorosa dell'informazione
 - non scienza del calcolatore né di Internet
- Sorella della **matematica** ... e della **filosofia**
 - radici storiche nella cultura classica ellenistica
 - importanti risultati teorici e di base all'inizio del 900
- Tuttavia

Informatica

- Grande impatto applicativo, industriale, e sociale
- Determinato dall'evoluzione tecnologica
 - elettronica
 - trasmissione (telecomunicazioni)
- Un moderno PC è enormemente più potente di un calcolatore da decine di miliardi degli anni 60
 - si va verso l'informatica “globale” e ... spesso “nascosta”



Anni 80...

- CPU: 1,01 Mhz
- RAM: 5-27 Kb
- Grafica: 176 x 184 16 colori
- Floppy disk da 170 Kb o nastri

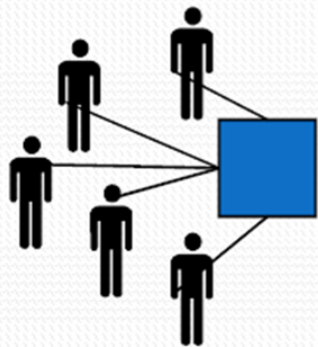


Il mio computer oggi

- CPU: 8-core Intel i7 3,10 Ghz
- RAM: 16 GB
- Grafica: dual monitor 1920x1080 + 2560x1440 milioni di colori
- Disco: 256 GByte SSD + 8TByte over NAS



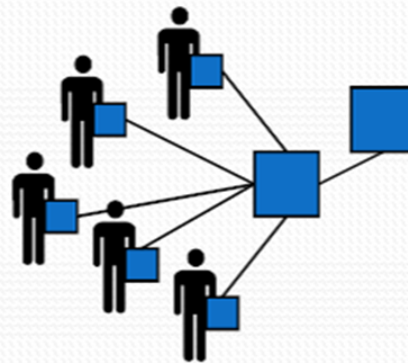
The evolution of information technology towards “ Pervasive Computing ”



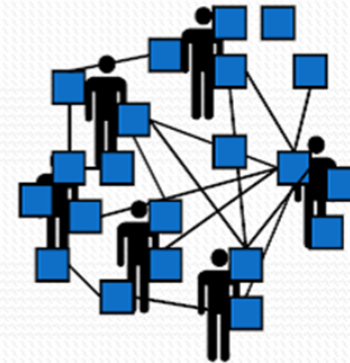
1950 : Mainframe



1980: Micro computer



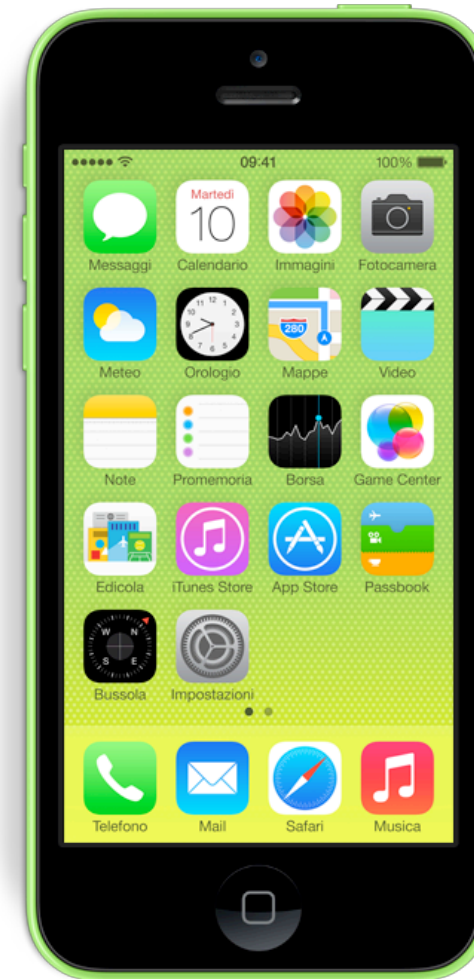
1990: Internet



200? Diffuse IT

Dispositivi mobili

- Processore dual core, con frequenza fino a 1.5 GHz
- Giroscopio a 3 assi
- Accelerometro
- Sensore di prossimità
- Sensore di luce ambientale
- GPS assistito
- Bussola digitale



Sensori e attuatori intelligenti

- Microcontrollori a 16-bit, 8 Mhz, 8 Kbyte RAM
- Interfacce wireless a bassa potenza
- Alimentazione a batteria
- Sensori e attuatori collegati alla board principale
 - tipicamente dettati dall'applicazione



Il software oggi

- Il **software** è parte essenziale di molti prodotti di largo consumo
 - dal telefonino alla lavatrice,
dall'automobile al forno
- Spesso il software non è il prodotto, ma è una parte del prodotto
 - deve essere ingegnerizzato con il resto dell'applicazione

Rappresentazione e manipolazione dell'informazione

Rappresentazione dell'informazione

- Il primo tipo di informazione che si presta ad essere rigorosamente rappresentato è **l'informazione numerica**
 - n aste per rappresentare il numero n
(numerazione unaria)
 - $\lfloor \log_k(n) \rfloor + 1$ cifre per rappresentare il numero n
in base k

Informazione non numerica

- Informazione **testuale** (caratteri)
- Informazione **grafica** (pixel, forme geometriche)
- Informazione **musicale** (digitalizzata o no)
- **Video**
- Rappresentazione dell'informazione in forma analogica e digitale

Algoritmo

- Codifichiamo l'informazione per poter applicare **algoritmi**
 - una sequenza di passi elementari, ben definita, precisa, eventualmente eseguibile anche da una macchina
- Informazione ed elaborazioni complesse vengono sempre scomposte in elementi base semplici e aggregate
- Questa è l'essenza della progettazione informatica!

Esseri umani e calcolatori

- Gli **esseri umani** sono buoni esecutori di algoritmi
 - possono anche decidere di abbandonarli usando il buon senso
- I calcolatori **non** possiedono buon senso e intuizione
 - tutte le situazioni fuori dal normale devono essere loro descritte, se si vogliono ottenere reazioni appropriate



Algoritmi

- Gli **algoritmi** dipendono dalla rappresentazione dei dati (informazioni) prescelte
 - rappresentazione ed elaborazione dell'informazione sono correlati
- Gli **algoritmi** possono essere più o meno **efficienti**
- Tale valutazione va tenuta presente, per ora, almeno sul piano intuitivo

Esempio: cammino minimo

Problema: trovare il percorso più corto tra due città c_p e c_a

- 1) Si trovano tutte le sequenze di città che determinano un itinerario tra le due città c_p e c_a
- 2) Per ogni sequenza, si calcola la somma delle distanze dei vari tratti di strada
- 3) Si individua la sequenza per cui il valore della somma delle distanze è minimo e la si sceglie come **strada più breve**
 - se per caso si trovasse più di un itinerario con la stessa distanza totale tra c_p e c_a , se ne sceglierebbe uno arbitrariamente
 - per esempio il primo trovato
 - se non si trovasse alcun itinerario, per esempio perché c_p e c_a sono separate dal mare, non esiste alcuna soluzione

Sottoproblemi

- Il punto 1 merita un esame più approfondito
 - trovare tutti gli itinerari che vanno da una città all'altra è a sua volta un problema che va risolto mediante un opportuno algoritmo
 - possiamo chiamarlo un **sotto-problema** del problema principale
 - la sua soluzione è necessaria per costruire la soluzione del problema principale
 - questo sotto-problema, a sua volta, può essere risolto mediante un algoritmo

Punto 1

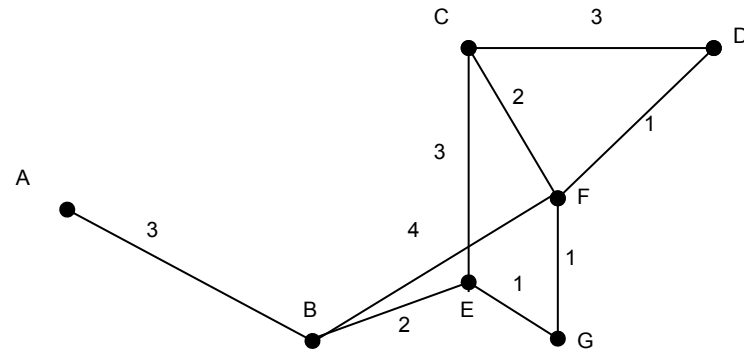
- Consideriamo come itinerario di lunghezza 0 un itinerario fittizio, che parte e arriva nella città c_p
- A questo punto, gli itinerari di lunghezza 1 saranno tutti quelli che collegano c_p alle città limitrofe, cioè a quelle città direttamente collegate a c_p da una strada

Punto 1

- Se n è il numero di città riprodotte nella carta geografica, un itinerario che unisca c_p a c_a senza passare due volte (inutilmente!) dalla stessa città, non può essere costituito da un numero di tratti elementari superiore a $n-1$
- Basta quindi costruire tutti gli itinerari che partono da c_p e hanno un numero di tratti non superiore a $n-1$, e scegliere tra questi quelli che terminano in c_a
- Supponiamo di aver già trovato tutti gli itinerari che partono da c_p e sono lunghi $r-1$ tratti
 - otterremo gli itinerari lunghi r aggiungendo a ciascuna copia un tratto ulteriore, che collega l'ultima città a tutte le città direttamente collegate a essa, purché tali città non facciano già parte della sequenza lunga $r-1$

Induzione

- Abbiamo così definito i primi due passi di un algoritmo, che in n passi ci porta a generare tutti i percorsi lunghi $n-1$ che partono da c_p , e quindi a risolvere il punto 1 dell'algoritmo generale
- Questo ragionamento è un primo esempio di un potentissimo procedimento matematico-informatico che ci servirà per risolvere i più svariati problemi: **l'induzione**
- Otteniamo dunque una nuova formulazione più completa e più precisa del punto 1 del nostro algoritmo



Città collegate ad A da una strada
 Città collegate a B da una strada
 Città collegate a C da una strada
 Città collegate a D da una strada
 Città collegate a E da una strada
 Città collegate a F da una strada
 Città collegate a G da una strada

{B}
 {A, E, F}
 {D, E, F}
 {C, F}
 {B, C, G}
 {B, C, D, G}
 {E, F}

Percorsi di 0 tratti che partono da E
 Percorsi di 1 tratto che partono da E
 Percorsi di 2 tratti che partono da E
 Percorsi di 3 tratti che partono da E
 Percorsi di 4 tratti che partono da E

{E}
 {E, B} {E, C} {E, G}
 {E, B, A} {E, B, F} **{E, C, D}** {E, C, F} {E, G, F}
 {E, B, F, C} **{E, B, F, D}** {E, B, F, G} {E, C, D, F} {E, C, F, B}
{E, C, F, D} {E, C, F, G} {E, G, F, B} {E, G, F, C} **{E, G, F, D}**
{E, B, F, C, D} {E, B, F, D, C} {E, C, D, F, B} {E, C, D, F, G}
 {E, C, F, B, A} {E, G, F, B, A} **{E, G, F, C, D}** {E, G, F, D, C}

Lunghezza del percorso {E, C, D} = 6
 Lunghezza del percorso {E, B, F, D} = 7
 Lunghezza del percorso {E, C, F, D} = 6
 Lunghezza del percorso **{E, G, F, D}** = **3**
 Lunghezza del percorso {E, B, F, C, D} = 11
 Lunghezza del percorso {E, G, F, C, D} = 7

Percorso più breve

Cosa abbiamo imparato?

- Certamente noi **non** consultiamo le carte geografiche in questa maniera
- Un problema complesso si può risolvere scomponendolo in **problemi meno complessi** fino ad arrivare a **problemi elementari**
- Quali siano i problemi elementari dipende da chi/che cosa è il risolutore
- Trovato un algoritmo non è detto che questo sia l'unico né tantomeno il migliore per risolvere il problema
- Quando potremo stabilire che l'algoritmo da noi elaborato è eseguibile da una macchina?