

Introduzione all'informatica

www.unidocs.it

www.unidocs.it

www.unidocs.it



www.unidocs.it

www.unidocs.it



www.unidocs.it

www.unidocs.it



www.unidocs.it

www.unidocs.it

Perché programmare

- Perché insegnare programmazione agli studenti di ingegneria?
- Per rispondere, cominciamo a capire che cosa è l'informatica
- Definizione di Ceri, Mandrioli, Sbatella:
“(L'informatica è) la scienza della rappresentazione e dell'elaborazione dell'informazione”
- Definizione dell' Association for Computing Machinery (ACM):
“L'informatica è lo studio sistematico degli algoritmi che **descrivono e trasformano** l'informazione: la loro teoria, analisi, progetto, efficienza, realizzazione, e applicazione”
- L'informatica **non è** la scienza del calcolatore, **non è** la scienza di internet, **non è** la scienza dei siti web...

Perché programmare (2)

- Una definizione piuttosto rudimentale di **“programmazione”** potrebbe essere:
“Far eseguire al calcolatore ciò che noi vogliamo che faccia”
- Mediante la programmazione noi diciamo al computer come rappresentare ed elaborare le informazioni
- Il computer per l’informatica è un mezzo (molto potente), ma non è il fine
 - volendo, si può fare informatica anche senza computer (in origine il “computer” era il “computatore”, ovvero un essere umano)
 - il computer però è molto più efficiente di un essere umano
- Quali informazioni possiamo rappresentare/elaborare? Di fatto qualunque:
 - fatture, carriera universitaria degli studenti, conto corrente
 - figure geometriche, edifici, ponti, terreni, immagini
 - genoma umano
 - ...
- Le informazioni che devono essere elaborate da un computer devono prima essere rappresentate in forma digitale

Rappresentazione digitale dell'informazione

- Caratteri
 - A ogni lettera (maiuscole e minuscole), cifra e simbolo (parentesi, punteggiatura, ...) si assegna un numero, secondo il codice ASCII (American Standard Code for Information Interchange), ad esempio.
- Immagini
 - L'immagine viene discretizzata in punti, quindi una o più unità di informazione numerica esprime per ogni punto il colore secondo una tabella o indicando le tre componenti RGB.
- Musica
 - Il suono degli strumenti viene "campionato" a intervalli regolari. Ogni campione contiene informazione numerica che rappresenta la forma d'onda corrispondente. Per i CD la frequenza di campionamento è di 44100 Hz.

Linguaggio di programmazione

- “Far eseguire al calcolatore ciò che noi vogliamo che faccia”
- Un **linguaggio di programmazione** è un modo per descrivere al computer le informazioni da elaborare e le operazioni da compiere
- Esistono diversi linguaggi di programmazione:
 - C
 - Python
 - C++
 - Java
 - Lisp
 - Prolog
 - ...
- I principi di base sono molto simili tra i vari linguaggi di programmazione, una volta imparati quelli, imparare un nuovo linguaggio di programmazione non è particolarmente difficile

quello che vedremo noi

a oggetti

funzionale

logico

Programmazione

- Il computer nasce come macchina per fare calcoli
- Il computer non è molto “intelligente” (non sa fare più di quello che gli viene detto di fare), ma quel che fa (i calcoli) lo fa molto bene
 - purchè gli venga detto correttamente come farli...
- Saper programmare significa saper far eseguire al computer i calcoli, le elaborazioni che noi vogliamo che vengano eseguite

La programmazione e gli algoritmi

- Programmare = far risolvere al computer un determinato problema
- Il primo, fondamentale passo però è di decidere come il problema va risolto
- Il “come” un problema viene risolto (in maniera meccanica, cioè eventualmente tramite un calcolatore) è definito mediante un **algoritmo**

Il concetto di algoritmo

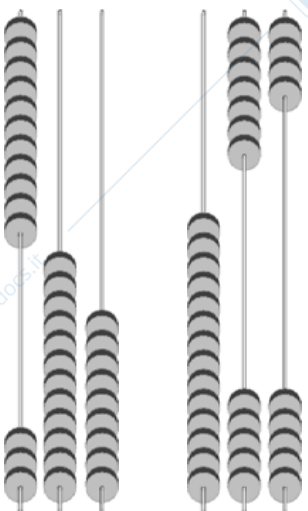
- **Definizione informale di algoritmo:**
una sequenza precisa e finita di operazioni, comprensibili da un esecutore, che portano allo svolgimento di un compito (cioè alla soluzione del problema considerato)
- Il termine “algoritmo” deriva dal nome del matematico persiano Muhammad ibn Musa 'l-Khwārizmi, che pubblicò, tra gli altri, il “Libro sulla ricomposizione e sulla riduzione”, dal quale prende le origini la parola Algebra
- **Alcune considerazioni:**
 - il concetto di algoritmo presuppone l’esistenza di un esecutore, cioè colui che deve effettivamente compiere le operazioni
 - l’esecutore è spesso, ma non sempre, il computer
 - le operazioni devono essere precise abbastanza per essere comprensibili dall’esecutore
 - il numero di operazioni deve essere finito
- **Esempi di algoritmi:**
 - istruzioni per la cottura della pasta
 - istruzioni per il montaggio di un mobile
 - calcolo del massimo comune divisore di numeri naturali
 - prelevamento di denaro tramite terminale Bancomat

Esempio di algoritmo

- Algoritmo di cottura degli spaghetti:
 1. Mettere l'acqua sul fuoco
 2. Attendere l'ebollizione
 3. Mettere il sale
 4. Buttare gli spaghetti
 5. Attendere x minuti
 6. Scolare la pasta
- Questo è un algoritmo puramente sequenziale, le operazioni vengono eseguite in un unico ordine, da 1 a 6
- Ci sono più modi di risolvere lo stesso problema, quindi ci sono più algoritmi che portano allo stesso risultato.
- Una semplice variazione dell'algoritmo di cottura degli spaghetti:
 1. Mettere l'acqua sul fuoco
 2. Mettere il sale
 3. Attendere l'ebollizione
 4. Buttare gli spaghetti
 5. Attendere x minuti
 6. Scolare la pasta

Un esempio di algoritmo di calcolo

- Somma di due numeri tramite pallottoliere



situazione iniziale

situazione finale

1. sposta la pallina da sinistra a destra nella prima riga, al contempo da destra a sinistra nella terza
2. ripeti l'operazione precedente fino a svuotare la parte sinistra della prima riga.
3. sposta la pallina da sinistra a destra nella seconda riga, al contempo da destra a sinistra nella terza
4. ripeti l'operazione precedente fino a svuotare la parte sinistra della seconda riga.

Algoritmi, programmi, programmatori

- Gli algoritmi sono descritti tramite **programmi**, cioè sequenze di istruzioni scritte in un linguaggio (linguaggio di programmazione) comprensibile al calcolatore
 - se l'esecutore è il computer, devo descrivergli ciò che deve fare (l'algoritmo) in un linguaggio che lui capisca
 - ===> linguaggio di programmazione
- Il compito del programmatore informatico è produrre algoritmi (cioè capire la sequenza di passi che portano alla soluzione di un problema) e codificarli in programmi (cioè renderli comprensibili al calcolatore)

Caratteristiche desiderate degli algoritmi

- **E' fondamentale** che l'algoritmo prodotto sia **corretto ed efficiente**
- Un algoritmo deve essere **corretto**
 - deve cioè risolvere il problema desiderato
 - un algoritmo che risolve un problema diverso da quello desiderato non serve a nulla
 - spesso lo sforzo di capire quale è il problema da risolvere non è banale
- Un algoritmo deve essere **efficiente**
 - deve cioè fare un buon uso delle risorse a disposizione (tempo, spazio, ecc.)
 - noi in questo corso non ci preoccuperemo di come scrivere algoritmi (e programmi) efficienti, ma l'efficienza è comunque un problema cruciale per chi scrive algoritmi

Esempio: gestione di una biblioteca

- **Descrizione del problema:**
 - Libri disposti sugli scaffali
 - Ogni libro si trova in una precisa e invariabile posizione con due coordinate: scaffale e posizione nello scaffale
 - La biblioteca è dotata di uno schedario (ordinato in ordine alfabetico per autore/i e titolo)
 - Ogni scheda contiene, nell'ordine:
 - cognome e nome dell'autore
 - titolo del libro
 - data di pubblicazione
 - numero dello scaffale in cui si trova
 - numero d'ordine della posizione attribuita al libro nello scaffale.
- **Algoritmo per prelevare un libro dalla biblioteca:**
 1. Decidi il libro da richiedere
 2. Preleva il libro richiesto
- Il secondo passo non è ancora elementare abbastanza, non è sufficientemente chiaro come vada realizzato.
- Servono maggiori dettagli

Gestione della biblioteca (2)

1. Decidi il libro da richiedere
 2. Cerca la scheda del libro richiesto
 3. Segnati numero scaffale e posizione
 4. Cerca lo scaffale indicato
 5. Accedi alla posizione indicata e preleva il libro
 6. Scrivi i tuoi dati sulla "scheda prestito"
- Questo algoritmo è già più preciso, ma potrebbe essere dettagliato ulteriormente
 - Il passo più complicato è certamente il 2: come faccio per "cercare la scheda del libro richiesto"?

Sottoalgoritmo

- Mi serve un nuovo algoritmo per risolvere il problema “dato l’autore e il titolo di un libro, trova la scheda corrispondente nello schedario”
 - mi serve un **sottoalgoritmo** per risolvere il passo 2
- Il primo algoritmo che probabilmente viene in mente:
 1. esamina la prima scheda dello schedario
 2. Se autore e titolo della scheda corrente sono quelli cercati allora stop, ricerca conclusa con successo, altrimenti passa alla scheda successiva
 3. Se sei in fondo allo schedario allora stop, la scheda cercata non esiste altrimenti passa alla scheda successiva
 4. ripeti dal passo 2.
- Questo algoritmo certamente funziona (controllo tutte le schede, prima o poi, se la scheda esiste, la trovo)

Considerazioni e miglioramenti

- Se cerco un libro di Asimov, con pochi passi di questo algoritmo di ricerca arrivo alla scheda desiderata, ma se cerco un libro di Zulueta da Costa?
- Si intuisce che l'algoritmo non è dei più efficienti...
- Per fare un algoritmo più efficiente possiamo sfruttare il fatto di sapere che lo schedario è **ordinato** per autore:
 1. prendi la scheda centrale dello schedario
 2. **se** è la scheda cercata la ricerca ha termine con successo
 3. **se** la scheda cercata segue alfabeticamente quella esaminata, ripeti la ricerca nella seconda metà dello schedario
altrimenti ripeti la ricerca nella prima metà
- Questo algoritmo però ha un problema...
 - non contempla il caso in cui la scheda cercata non esista
 - più precisamente, il caso in cui ad un certo punto la parte di schedario in cui fare la ricerca sia vuota

Ulteriore modifica

- Modifica che lo rende corretto anche nel caso in cui la scheda non esista:
 1. se lo schedario è vuoto allora termina con **insuccesso altrimenti** prendi la scheda centrale dello schedario
- Ora l'algoritmo non solo è corretto, ma è anche "più efficiente" di quello precedente
 - In realtà, a guardarlo bene, ci impiega un po' di più a trovare il libro di Asimov, ma molto di meno a trovare quello di Zulueta da Costa...
 - ...ma sono considerazioni che non approfondiremo ulteriormente

Un algoritmo numerico

- Problema: calcolo del massimo comun divisore (MCD) di due numeri naturali
- Algoritmo banale:
 1. Prendi il minore tra i due numeri
 2. Se li divide entrambi allora stop, è il MCD
 3. altrimenti, passa al numero immediatamente inferiore
 4. ripeti dal passo 2
- Un algoritmo più efficiente è quello inventato da Euclide:
 1. Se il numero più piccolo dei due divide esattamente (cioè con resto 0) il più grande allora stop, quello è il MCD.
 2. altrimenti, calcola il resto della divisione dei due numeri, e ripeti dal passo 2 usando come numeri il resto ed il minore dei numeri di partenza
- Usiamo l'algoritmo per calcolare il MCD di 2079 e 987:
 - dividiamo 2079 per 987 e otteniamo quoziente=2 e resto=105
 - dividiamo 987 per 105 e otteniamo quoziente=9 e resto=42
 - dividiamo 105 per 42 e otteniamo quoziente=2 e resto=21
 - dividiamo 42 per 21 e otteniamo quoziente=2 e resto=0– siamo arrivati a resto nullo: l'ultimo divisore (21) è il MCD che stiamo cercando

Il concetto di variabile

- Ad ogni ripetizione dell'algoritmo ci sono 4 numeri in gioco: i due numeri di cui si vuole il MCD, il quoziente, ed il resto:
 - dividiamo 2079 per 987 e otteniamo quoziente=2 e resto=105
 - dividiamo 987 per 105 e otteniamo quoziente=9 e resto=42
 - ...
 - dividiamo \square per \square e otteniamo quoziente= \square e resto= \square
- Diamo un nome alle 4 “caselle” per distinguerle una dall'altra:
 - dividiamo x per y e otteniamo quoziente=q e resto=r
- X, Y, q, r è come se fossero dei “contenitori” in cui noi mettiamo dei valori
- quando passeremo a parlare di linguaggi di programmazione (e di C in particolare) questi “contenitori” li chiameremo “variabili”
 - attenzione!!! *Non* è il è il concetto matematico di “variabile”, ma qualcosa di diverso (vedremo meglio)
- In questi “contenitori” posso mettere dei valori, per esempio potrei mettere il valore 2079 nel contenitore x, il contenuto del contenitore r nel contenitore y
 - $x \leftarrow 2079$
 - $y \leftarrow r$
- Potrei anche fare cose “più complicate”, per esempio mettere nel contenitore q il valore della divisione di ciò che sta in x per ciò che sta in y:
 - $q \leftarrow x/y$

riscriviamo l'algoritmo...

- Obiettivo: dati due numeri non nulli a e b (con $a > b$) trovare il loro MCD
 1. metti il numero a nel contenitore x
 2. metti il numero b nel contenitore y
 3. dividi il contenuto di x per il contenuto di y ; metti il quoziente in q , ed il resto in r
 4. se r è uguale a 0 , allora stop, y contiene il MCD cercato
 5. altrimenti, metti in x il valore che c'è in y , ed in y il valore che c'è in r
 6. ripeti dal passo 3
- Riscriviamo ulteriormente l'algoritmo
 1. $x \leftarrow a$
 2. $y \leftarrow b$
 3. $q \leftarrow x/y, r \leftarrow x - q * y$
 4. se $r = 0$, allora stop, y è il MCD
 5. altrimenti $x \leftarrow y, y \leftarrow r$
 6. ripeti dal passo 3

Notazione: diagrammi di flusso

