

# IN1 Informatica Generale (1<sup>o</sup> Modulo)

A.A. 1998/1999

Marco Liverani

## 1. Problemi ed algoritmi

Introduzione alle caratteristiche del calcolatore ed al rapporto programmatore/esecutore; compiti ed abilità del programmatore; principali caratteristiche ed abilità dell'esecutore, operazioni di base (logiche, aritmetiche e di confronto).

Linguaggi di programmazione. Istruzioni fondamentali di un linguaggio di programmazione generico. Algoritmi e programmi; diagrammi di flusso. Regole della programmazione strutturata; approccio *top-down* alla soluzione di un problema.

## 2. Architettura del calcolatore

Struttura interna di un calcolatore; processo di esecuzione di un programma. Architettura esterna: periferiche, unità di memoria di massa, ecc. CPU, processori RISC e CISC; calcolatori paralleli.

Evoluzione recente dell'informatica: anni 50-60: *mainframe*; anni 70: *minicomputer*, il sistema operativo UNIX; anni 80: *microcomputer*, *personal computer*; anni 90: *personal computer*, *workstation*, collegamento in LAN, architettura *client/server*.

Sistemi operativi: funzioni di base; sistemi operativi multiutente *multitasking* e *multithreadmin*. Sistemi operativi proprietari e aperti; i servizi, i demoni; le funzioni relative alla sicurezza. *Shell* di comandi ed interfaccia utente grafica (GUI). Reti di calcolatori; sistemi operativi di rete; protocolli di rete.

## 3. Il linguaggio C

Organizzazione della memoria di un calcolatore, indirizzi, parole, puntatori. Codifica binaria ed esadecimale. Tipi di dato (interi, float, double, char, stringa). Strutture dati (array, matrici, liste, alberi, grafi).

Linguaggio macchina, linguaggi di alto livello; compilatori ed interpreti. Il linguaggio C: scopi e principali caratteristiche.

La struttura di un programma C, l'inclusione degli *header*, dichiarazione delle variabili; le librerie.

Tipi di dato elementari in linguaggio C: interi, *floating point*, *double*, *char*. Puntatori; aritmetica sui puntatori. *Array* e matrici e loro rappresentazione in memoria. Strutture dati complesse: liste, alberi, grafi; l'istruzione "**struct**".

Strutture di controllo: "**if ... else ...**", "**while ...**", "**do ... while**", "**for ...**". Operatore di assegnazione, operatori aritmetici in C in forma estesa e compatta.

Funzioni: funzioni di libreria e funzioni definite dall'utente. Passaggio di parametri per valore e per indirizzo alle funzioni. Funzioni ricorsive.

Funzioni di input/output: “`printf`”, “`scanf`”; funzioni per la gestione della memoria: “`malloc`”, “`free`”, “`sizeof`”.

#### 4. Algoritmi fondamentali

Algoritmi di ordinamento: *insertion sort*, *selection sort*, *bubble sort*, *quick sort*; algoritmi ottimi: *heap sort*, *merge sort*. Complessità di un algoritmo, analisi della complessità degli algoritmi di ordinamento.

Algoritmi su grafi. Definizioni principali: grafo, grafo orientato, grafo connesso; sottografo, sottografo indotto; passeggiata, cammino, ciclo; *cut-set*; lista, albero, *spanning tree* di un grafo. Strutture dati per la rappresentazione di grafi mediante un calcolatore: liste di adiacenza e matrici di adiacenza. Algoritmi di visita di un grafo: visita in ampiezza (BFS), visita in profondità (DFS). Il problema del minimo albero ricoprente: l'algoritmo di Kruskal.

Cenni sulla teoria della complessità: problema di ottimizzazione e problema in forma decisionale. La classe dei problemi polinomiali (P). La classe dei problemi NP, algoritmi di verifica. La classe dei problemi NP-completi. Il problema “P=NP”.

## TESTI CONSIGLIATI

- [1] T.H. CORMEN, C.E. LEISERSON, R.L. RIVEST, *Introduzione agli algoritmi*. Jackson Libri, (1997).
- [2] B.W. KERNIGHAN, D.M. RITCHIE, *Linguaggio C*. Gruppo Editoriale Jackson, (1985).
- [3] H. SCHILDT, *Linguaggio C - La guida completa*. McGraw-Hill, (1995).

## BIBLIOGRAFIA SUPPLEMENTARE

- [4] G. AUSIELLO, A. MARCHETTI-SPACCAMELA, M. PROTASI, *Teoria e progetto di algoritmi fondamentali*. Franco Angeli, (1988).

## MODALITÀ D'ESAME

- valutazione in itinere (“esoneri”)		<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO
- esame finale	scritto	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO
	orale	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO
- altre prove di valutazione del profitto (meglio descritte sotto)		<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO

L'esame si articola su tre fasi distinte:

1. una prova scritta;
2. una prova orale;
3. la discussione orale di una tesina facoltativa.

Per accedere alla prova orale è necessario superare con un voto sufficiente la prova scritta, o le due prove di esonero previste. La prova scritta consiste nella soluzione di esercizi: viene richiesto di presentare una codifica in linguaggio C di un algoritmo che risolva il problema dato. La prova orale consiste nella discussione di alcuni tra i principali argomenti trattati durante il corso. Per migliorare il giudizio finale è possibile presentare una tesina facoltativa: tale tesina consiste nella discussione di un argomento che sarà stato preventivamente assegnato allo studente e nella presentazione di una codifica in linguaggio C di uno degli algoritmi presentati nella trattazione teorica. Gli argomenti delle tesine saranno di livello equivalente agli ultimi argomenti trattati nel programma del corso. Le tesine vengono assegnate dal docente su richiesta dei singoli studenti che decidono di presentare anche la tesina facoltativa a completamento della prova orale.