

Algoritmi (modulo di laboratorio)

Corso di Laurea in Matematica

Roberto Cordone

DI - Università degli Studi di Milano



- Lezioni: Martedì 8.30 - 10.30 in aula 8 Mercoledì 10.30 - 13.30 in aula 309
Giovedì 16.30 - 18.30 in aula 307 Venerdì 10.30 - 12.30 in aula 4
- Ricevimento: su appuntamento (Dipartimento di Informatica)
- E-mail: roberto.cordone@unimi.it
- Pagina web: <http://homes.di.unimi.it/~cordone/courses/2024-algo/2024-algo.html>
- Sito Ariel: <https://mgoldwurmasd.ariel.ctu.unimi.it>

Obiettivi e struttura del laboratorio

Il modulo di laboratorio si propone di

- **descrivere in dettaglio alcuni algoritmi e strutture dati** introdotti nelle lezioni teoriche
- **legare le nozioni generali e astratte agli aspetti più tecnici e concreti**
- **realizzare in pratica gli algoritmi in un linguaggio di programmazione** (nel caso specifico, il C)

NON È UN CORSO DI PROGRAMMAZIONE!

Il modulo consiste in 48 ore che si sviluppano attraverso

- 11 lezioni da 3 ore
- una pausa per Pasqua
- 3 lezioni da 3 ore
- una pausa per esami
- 2 lezioni da 3 ore

Modalità d'esame (1)

L'esame si compone di due parti successive

- 1 presentazione di un **progetto** per il modulo di laboratorio
- 2 **prova orale** per il modulo di teoria

Il progetto cambia ad ogni appello ed è lo stesso per tutti gli iscritti

Richiede di

- **modellare un problema** “concreto”
- **progettare un algoritmo** per risolvere il modello
- **scrivere un programma** in C che realizza l'algoritmo
- **stendere una relazione** che descriva problema e algoritmo

Testo e data di scadenza sono pubblicati sulla pagina web del laboratorio

C'è anche un file di indicazioni utili dettagliate

Modalità d'esame (2)

Entro la data di scadenza occorre spedire al docente di laboratorio

- il codice (in C) adeguatamente commentato
- una relazione (in PDF) che descriva:
 - ① problema: il problema concreto proposto che si intende risolvere
 - ② modello: scelta motivata degli oggetti matematici usati per descrivere il problema
 - ③ algoritmo: descrizione dell'algoritmo risolutivo e analisi della sua complessità computazionale (tempo e spazio)

NON È UN MANUALE TECNICO!

La valutazione terrà conto di:

- ① **relazione:**
 - a **completezza**, cioè descrizione di ogni aspetto rilevante del progetto
 - b **correttezza**, cioè corrispondenza della relazione con teoria e codice
 - c **struttura**, cioè organizzazione e aspetto estetico
- ② **codice:**
 - a **correttezza**, cioè assenza di errori sintattici e semantici
 - b **efficienza**, cioè uso di strutture dati e algoritmi appropriati
 - c **struttura**, cioè organizzazione in moduli, funzioni, blocchi, portabilità e adeguatezza dei commenti

- Algoritmi e programmi
- Definizione e valutazione di complessità asintotica
- Richiami di programmazione in C
- Strutture dati e algoritmi elementari:
 - vettori, tabelle, stringhe e *record*
 - liste concatenate
 - algoritmi di ordinamento
- Strutture dati e algoritmi avanzati:
 - grafi: algoritmi di visita e componenti connesse
 - alberi di ricerca: algoritmi di costruzione e modifica
 - strutture per *union/find*: gestione di partizioni
 - code con priorità
 - algoritmi di *divide-et-impera*
 - algoritmi di programmazione dinamica
 - algoritmi *greedy*

Materiali di riferimento (1)

Per quanto riguarda gli **algoritmi** e le **strutture dati**:

- **dispense dal sito del docente di teoria**

[http://users.mat.unimi.it/users/goldwurm/Algoritmi\(Matematica\)/](http://users.mat.unimi.it/users/goldwurm/Algoritmi(Matematica)/)

- **lucidi dal sito del docente di laboratorio**

<https://homes.di.unimi.it/cordone/courses/2023-algo/2023-algo.html>

Ovviamente, è più semplice una ricerca con nome e cognome

Testi utili per consultazione:

- T. Cormen, C. Leiserson, R. Rivest, C. Stein, *Introduzione agli algoritmi e strutture dati*, McGraw-Hill, 2010.
(in inglese, *Introduction to algorithms*, 3rd edition)
- A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, 1974.
- C. Demetrescu, I. Finocchi, G. Italiano, *Algoritmi e strutture dati*, McGraw-Hill, 2004.
- A. Bertossi, *Algoritmi e strutture dati*, UTET Libreria, 2000.

Materiali di riferimento (2)

Per la **programmazione in C**, ogni testo o materiale va bene

Esempi (reperibili in rete):

- lucidi ed esercizi dal sito del docente di laboratorio
<https://homes.di.unimi.it/cordone/courses/2015-prog/2015-prog.html>
- K. N. King, *Programmazione in C*, W. W. Norton & Company, 2008.
(in inglese, *C Programming: A Modern Approach*, 2nd edition)
- A. Kelley, I. Pohl, *C. Didattica e Programmazione*, Pearson, Italia, 2004. (in inglese, *A book on C*, 4th edition)
- B. W. Kernighan, D. M. Ritchie, *Linguaggio C*, Pearson, Italia, 2004. (in inglese, *The C Programming Language*, 2nd edition)

È sufficiente un **compilatore C**

- in ambiente Linux, gcc è installato su tutte le distribuzioni
- in ambiente Windows,
 - durante le lezioni userò Dev-C++ senza interfaccia
 - per valutare il progetto userò Visual Studio Express
 - MinGW include una versione (datata) di gcc
<https://www.cs.colorado.edu/~main/bgi/cs1300/>

Le versioni con interfaccia grafica sono consigliabili solo a utenti esperti

Durante il corso non le useremo (ovviamente non sono vietate)