

Algoritmi (modulo di laboratorio)

Corso di Laurea in Matematica

Roberto Cordone

DI - Università degli Studi di Milano



Lezioni: Martedì 9.30 - 11.30 in aula Z5 **Mercoledì 11.30 - 13.30 in aula 2 e Z5**
 Giovedì 14.30 - 17.30 in aula 2 e Z5 Venerdì 11.30 - 13.30 in aula Z5

Ricevimento: **su appuntamento** (Dipartimento di Informatica)

E-mail: **roberto.cordone@unimi.it**

Pagina web: **<http://homes.di.unimi.it/~cordone/courses/2021-algo/2021-algo.html>**

Sito Ariel: **<https://mgoldwurma.ariel.ctu.unimi.it>**

Obiettivi e struttura del laboratorio

Questo modulo 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)

Il modulo consiste in

- 16 lezioni alternativamente da 2 e da 3 ore

che per l'emergenza in atto verranno

- trasmesse online in modo sincrono
- videoregistrate e rese disponibili sul sito Ariel del corso

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**

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 modello: descrizione motivata degli oggetti matematici usati per descrivere il problema
 - ② algoritmo: descrizione e pseudocodice dell'algoritmo risolutivo, analisi della sua complessità computazionale (tempo e spazio)
 - ③ programma: aspetti tecnici legati al linguaggio (solo se importanti)

La valutazione terrà conto di:

- ① **relazione:**
 - a **completezza**, cioè descrizione di ogni aspetto rilevante del progetto
 - b **correttezza**, cioè corrispondenza fra teoria, relazione 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, 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/2021-algo/2021-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,
 - MinGW include una versione (datata) di gcc
`https://www.cs.colorado.edu/~main/bgi/cs1300/`
 - durante le lezioni userò Dev-C++ senza interfaccia
 - per valutare il progetto userò Visual Studio Express

Le versioni con interfaccia grafica sono consigliabili solo a utenti esperti

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