# Algoritmi e strutture dati

## Prof. Marco L. Della Vedova

***OBIETTIVO DEL CORSO E RISULTATI DI APPRENDIMENTO ATTESI***

L’insegnamento si propone di fornire agli studenti le conoscenze, la mentalità e gli strumenti per l’analisi e la progettazione di algoritmi e strutture dati, con particolare riferimento al linguaggio Python.

Al termine dell’insegnamento lo studente sarà in grado di comprendere e analizzare algoritmi e strutture dati esistenti, sviluppare e implementare algoritmi originali per la risoluzione efficiente di problemi pratici.

***PROGRAMMA DEL CORSO***

* Introduzione: il ruolo degli algoritmi nell’informatica
* Modelli di calcolo e metodologie di analisi della complessità degli algoritmi
* Strutture dati: array, liste concatenate, pile, code, dizionari, *hash tables*, *heap*, alberi e grafi
* Analisi algoritmi di ordinamento
* Tecniche di programmazione: divide-et-impera, *backtracking*, programmazione dinamica, algoritmi *greedy*, algoritmi non deterministici
* Cenni di teoria dei grafi e algoritmi fondamentali sui grafi: ricerca in profondità, ricerca in ampiezza, componenti connesse, ordinamento topologico, *minimum spanning tree*, cammini minimi.

***BIBLIOGRAFIA***

Appunti delle lezioni e materiale didattico online.

Il manuale di riferimento è il possente

* *Introduction to Algorithms, Third Edition* di Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest e Clifford Stein, MIT-Press.

Un *must have* nella libreria di un vero informatico, è usato nelle più prestigiose università di tutto il mondo. Questo libro copre molti più argomenti rispetto a quelli trattati in questo corso.

Esiste anche la versione italiana dal titolo *Introduzione agli algoritmi e alle strutture dati*, edito dalla McGraw-Hill. Un altro valido testo è il C. Demetrescu, I. Finocchi, G. F. Italiano, *Algoritmi e strutture dati 2/ed*, McGraw-Hill. Una lettura di approfondimento è *Algorithms to live by - The computer science of human decisions*, di Brian Christian and Tom Griffiths.

***DIDATTICA DEL CORSO***

Lezioni in aula, lavori guidati in laboratorio e revisioni esercizi individuali.

***METODO E CRITERI DI VALUTAZIONE***

La prova d’esame consiste di due parti: un test scritto e una prova pratica di programmazione. Il test scritto è composto da domande sulla parte teorica del corso e/o da semplici esercizi; non è possibile consultare libri di testo o appunti durante lo svolgimento. La prova pratica di programmazione riguarda la risoluzione di uno o più problemi attraverso un programma in Python e si svolge al computer, a libri aperti con accesso a Internet. In entrambe le prove l’elaborato è personale e non è consentito il lavoro di gruppo. Test scritto e prova di programmazione si svolgono in un unico giorno con quattro ore di tempo a disposizione.

Il voto finale dell’esame è dato dalla somma dei punti ottenuti nella parte scritta (max 16), dei punti ottenuti nella prova di programmazione (max 15) e dei punti bonus ottenuti con gli homework.

***AVVERTENZE E PREREQUISITI***

È richiesta una conoscenza di base di programmazione. Se lo studente non conosce Python o le basi della programmazione, può comunque riuscire a seguire il corso normalmente, recuperando i concetti di base nelle prime settimane del corso con letture e attività consigliate dal docente.

*Orario e luogo di ricevimento degli studenti*

Ufficio del docente o da remoto. L’orario di ricevimento è consultabile alla pagina personale docente.