



UNIVERSITÀ  
degli STUDI  
di CATANIA

DIPARTIMENTO DI  
MATEMATICA E INFORMATICA

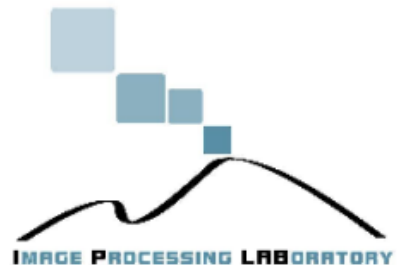
# Corso di Programmazione II

Alessandro Ortis

Image Processing Lab - [iplab.dmi.unict.it](http://iplab.dmi.unict.it)

[ortis@dmf.unict.it](mailto:ortis@dmf.unict.it)

[www.dmf.unict.it/ortis/](http://www.dmf.unict.it/ortis/)



# Obiettivi Formativi

- Acquisire **metodi** di programmazione
  - Imparare a ragionare da informatici
- Acquisire e sviluppare capacità di
  - Comprendere le proprietà fondamentali di diversi algoritmi e strutture dati per essi
  - Implementare gli algoritmi studiati in C++ in modo da ottenere soluzioni affidabili ed efficienti

# Perché “Efficienti”?

- I computer diventano sempre più rapidi e potenti perché curarsi di scrivere programmi efficienti?
- Perché l'efficienza è più importante della potenza

# Tempo $n^5$ (1 operaz. al secondo)

Dimensione dell'input	2	5	10	20	30	40	70
Tempo di calcolo	32	52	28	37	9	39	54
	Secondi	minuti	ore	giorni	mesi	mesi	anni

- Portiamo a 10 op/sec la potenza di calcolo

Dimensione dell'input	2	5	10	20	30	40	70
Tempo di calcolo	3,2	5,2	2,8	3,7	28	3,9	5,4
	Secondi	minuti	ore	giorni	giorni	mesi	anni

# Tempo $n^2$ (1 operaz. al secondo)

Dimensione dell'input	2	5	10	20	30	40	70
Tempo di calcolo	4 sec	25 sec	100 sec	7 min	15 min	27 min	82 min

- Meglio del metodo precedente implementato su una macchina 10 volte più veloce!

Dimensione dell'input	2	5	10	20	30	40	70
Tempo di calcolo	3,2 Secondi	5,2 minuti	2,8 ore	3,7 giorni	28 giorni	3,9 mesi	5,4 anni

# Attività formative

- 72 ore di lezione frontale (teoria+pratica)
- L'acquisizione dei metodi e delle competenze professionali della materia è sostenuta da
  - Frequenza delle lezioni
  - Studio individuale (tanto)
  - Studio del testo
  - Partecipazione al tutorato

# Testi

Luis Joyanes Aguilar

**Fondamenti di Programmazione in C++**

McGraw-Hill

T. H Cormen

**Introduction to Algorithms**

The MIT Press

Consigliati:

**C++ Tecniche avanzate di programmazione** Apogeo

**Algoritmi in C++** (terza edizione) Pearson Education Italia

**Effective C++ e More Effective C++** Addison-Wesley

# Tutorato

- 36 ore
- Orari prestabiliti (si consulti aulario)



# Relazione con altri insegnamenti

Propedeuticità richieste: per sostenere l'esame di Programmazione 2, dovrete aver **già superato**:

- Programmazione 1
- Strutture Discrete

# Programma delle lezioni - I

## **Parte 1: Cenni di complessità e tecniche ricorsive**

- Cenni di complessità asintotica
- Algoritmi elementari di ordinamento e ricerca
- Ricorsione ed algoritmi di ordinamento e ricerca basati sulla ricorsione

# Programma delle lezioni - II

## **Parte 2: Programmazione avanzata a oggetti - ripasso ed approfondimento**

- Classi e Oggetti
- Classi derivate
- Templates
- Sovraccaricamento degli operatori

# Programma delle lezioni - III

## **Parte 3: Strutture Dati**

- Liste
- Pile e code
- Alberi
- Grafi

## **Parte 1, 2 e 3:**

- Tantissimi esercizi

# Kahoot!



<https://kahoot.com/>

# Modalità d'esame



```
graph LR; A[Prova Scritta] --> B[Prova Pratica]; B --> C[Orale];
```

Prova  
Scritta

Prova  
Pratica

Orale

# Modalità d'esame



## 1. Prova scritta:

- Durata 1h
- 30 quesiti a risposta multipla
- Tutti i fogli utilizzati vanno consegnati
- Il voto dello scritto può essere conservato, solo su richiesta, al massimo nella stessa sessione

# Modalità d'esame



2. Prova pratica (assegnazione via mail @studium.unict.it):

- Realizzazione di un progetto software
- Consegna tramite repository **github**
- Compilazione mediante g++ su una macchina generica



# Modalità d'esame



## 3. Esame Orale:

- Compilazione ed esecuzione del progetto
- Quesiti sull'implementazione
- Quesiti su tutto il programma didattico

Domande?