

# MATEMATICA (LM39)

(Lecce - Università degli Studi)

## Insegnamento **CALCOLABILITA' E COMPLESSITA' COMPUTAZIONALE**

GenCod A004925

**Docente titolare** ANTONIO MARIO CARUSO

**Insegnamento** CALCOLABILITA' E COMPLESSITA' COMPUTAZIONALE

**Insegnamento in inglese** CALCULABILITY AND COMPUTATIONAL

**Settore disciplinare** INF/01

**Corso di studi di riferimento** MATEMATICA

**Tipo corso di studi** Laurea Magistrale

**Crediti** 6.0

**Ripartizione oraria** Ore Attività frontale: 42.0

**Per immatricolati nel** 2018/2019

**Erogato nel** 2019/2020

**Anno di corso** 2

**Lingua** ITALIANO

**Percorso** GENERALE

**Sede** Lecce

**Periodo** Primo Semestre

**Tipo esame** Orale

**Valutazione** Voto Finale

**Orario dell'insegnamento**

<https://easyroom.unisalento.it/Orario>

### BREVE DESCRIZIONE DEL CORSO

Lo scopo del corso è l'acquisizione di competenze e conoscenze di base dell'Informatica Teorica: il corso introdurrà le nozioni formali relative al concetto di funzione calcolabile da parte di una macchina di Turing, il problema della fermata e la riduzione tra linguaggi. Per la parte di complessità computazionale si definiranno e studieranno le proprietà delle classi: P, NP, NP-C e Np-Hard, ed il concetto di riduzione polinomiale. Enunciato e cenni della dimostrazione del Teorema di Cook-Levin. Si utilizzerà il linguaggio Python per studiare alcuni problemi Np-Hard come Zaino e Commesso Viaggiatore.

### PREREQUISITI

Programmazione, Algoritmi.

### OBIETTIVI FORMATIVI

**Conoscenze e comprensione.** Possedere una preparazione di base sui concetti teorici relativi alla calcolabilità e alla complessità computazionale.

**Capacità di applicare conoscenze e comprensione:** # essere in grado di produrre semplici dimostrazioni rigorose di risultati matematici non identici a quelli già conosciuti, ma chiaramente correlati ad essi, # essere in grado di formalizzare matematicamente problemi relativi alla calcolabilità o complessità di funzioni/algoritmi di moderata difficoltà, in modo da facilitare la loro analisi e risoluzione, # essere capaci di leggere e comprendere, in modo autonomo, testi avanzati o articoli di rivista relativi a questi settori.

**Autonomia di giudizio.** L'esposizione dei contenuti e delle argomentazioni sarà svolta in modo da migliorare la capacità dello studente di riconoscere dimostrazioni rigorose e individuare ragionamenti fallaci.

**Abilità comunicative.** La presentazione degli argomenti sarà svolta in modo da consentire l'acquisizione di una buona capacità di comunicare problemi, idee e soluzioni riguardanti l'Informatica Teorica, sia in forma scritta che orale.

**Capacità di apprendimento.** Saranno indicati argomenti da approfondire, strettamente correlati con l'insegnamento, al fine di stimolare la capacità di apprendimento autonomo dello studente.

### METODI DIDATTICI

Lezioni frontali ed esercitazioni in aula

---

## MODALITA' D'ESAME

orale.

---

## ALTRE INFORMAZIONI UTILI

Pagina del Corso: <http://bilbo.unisalento.it/antonio/didattica/algoritmi-e-complessita/>

---

## PROGRAMMA ESTESO

**Calcolabilità:** Macchine di Turing, Riconoscimento vs calcolo di funzione. Problema della Fermata, linguaggi R, RE, co-RE. Riduzioni. (2 lezioni)

**Python:** Introduzione, parte imperativa, parte funzionale, parte ad oggetti. (6 lezioni)

**Complessità e Algoritmi:** Classi di Complessità: DTIME, NDTIME, PSPACE, NPSPACE. P vs NP. Definizioni diverse per NP e loro relazioni. NPSPACE, EXP, etc. Problemi Np-Completi e NP-Ardui, Teorema di Cook-Levin, SAT e riduzioni polinomiali, esempi di varie riduzioni. Limiti e Utilità della teoria della complessità computazionale. Algoritmi di Approssimazione, esempi vari. Algoritmi Probabilistici, Max-SAT, Matching, etc. Effetto soglia sulle istanze di SAT. (7 lezioni)

Esempi in Python di soluzioni per problemi Np-Completi.

---

## TESTI DI RIFERIMENTO

Quasi tutti i testi sotto sono reperibili liberamente come PDF su web. Usare un motore di ricerca per trovarli.

Italiano:

- Crescenzi, Informatica Teorica.
- Ausiello, D'Amore, Gambosi, Linguaggi, Modelli, Complessità.
- Asperti, Teoria della Calcolabilità.
- Vigna, Dispense per il corso di informatica Teorica.
- Degano, Calcolabilità.
- **Dovier, Giacobazzi, Linguaggi Formali, Calcolabilità e Complessità**

Inglese:

- Stephen G. Simpson, Foundation of Mathematics.
- Rogers, Theory of Recursive Functions.
- Crescenzi, Bovet, Introduction to the theory of Complexity.
- Bjorn Poonen, Indecidable Problems: A Sampler.