

# Fondamenti di Programmazione con Laboratorio CdL in MATEMATICA

## I Appello 24 Gennaio 2019

**N.B.:** Negli esercizi di programmazione, vengono valutati anche l'uso dei costrutti appropriati, l'uso delle condizioni booleane, la leggibilità e l'efficienza del codice proposto. Inoltre, non è consentito l'uso di variabili globali o di variabili statiche. Laddove è utilizzato, il tipo `boolean` è definito da `typedef enum {false, true} boolean`.

### ESERCIZIO 1

Sia  $L$  il linguaggio regolare su  $\Sigma = \{1, 2, 3\}$  tale che le stringhe  $w$  di  $L$ :

- cominciano sempre per 1
- non contengono mai le sequenze 21, 22, 31, 33;
- finiscono sempre per 3.

Si forniscano allora:

- l'automa a stati finiti **non deterministico**  $A$  in grado di riconoscere  $L$ ;
- la corrispondente **espressione regolare**, applicando l'algoritmo di eliminazione degli stati all'automa  $A$ ;
- la corrispondente **grammatica regolare**, usando l'algoritmo per passare dalle transizioni dell'automa alle produzioni della grammatica.

### ESERCIZIO 2

Dire se i seguenti linguaggi sono regolari o liberi dal contesto, giustificando la risposta e usando, se necessario, il pumping lemma per i linguaggi regolari e il pumping lemma per i linguaggi liberi.

- $L_1 = \{0^i 1^j 2^j 3^i \mid i \geq 1, j \geq 1\}$
- $L_2 = \{0^i 1^j 2^i 3^j \mid i \geq 1, j \geq 1\}$

### ESERCIZIO 3

Un vettore (contenente elementi di un insieme su cui è definita una nozione di ordinamento) è detto *unimodulare* se ha tutti valori distinti ed esiste un indice  $h$  tale che  $A[0] > A[1] > \dots > A[h-1] > A[h]$  e  $A[h] < A[h+1] < A[h+2] < \dots < A[n-1]$ , dove  $n$  è la dimensione del vettore.

Si scriva in C una funzione **ricorsiva** (non occorre scrivere il `main`) che, dato un vettore unimodulare  $A$  di  $n$  interi, e dati il suo primo e ultimo indice, restituisca il valore minimo del vettore.

La complessità in tempo dell'algoritmo deve essere  $O(\log n)$ .

### ESERCIZIO 4

- Si descriva a parole, in non più di 10 righe, il funzionamento dell'algoritmo MERGESORT, **oppure** se ne scriva il codice omettendo quello della funzione MERGE che può essere descritta a parole in non più di 4 righe.
- Si scriva di conseguenza la relazione di ricorrenza che descrive la complessità in tempo di MERGESORT.

\* Se ne indichi la soluzione, motivando la risposta usando uno qualsiasi dei metodi visti a lezione.

### ESERCIZIO 5

Dato un linguaggio  $L$  si definisce  $prefix(L) = \{w \mid \exists x \text{ t.c. } wx \in L\}$ . Notare che  $x$  può essere  $\epsilon$ . Ad esempio, se  $L = \{01, 101\}$  allora  $prefix(L) = \{\epsilon, 0, 01, 1, 10, 101\}$ .

Dimostrare che se  $L$  è un qualsiasi linguaggio regolare, allora  $prefix(L)$  è regolare. (Partire da un DFA per  $L$  e costruire un DFA per  $prefix(L)$ ).