

# Esercitazione 14

## Liste & Alberi

Alberto Marchesi  
Informatica A – Ingegneria Matematica (M—Z)

3 Dicembre 2021

Gli esercizi visti a lezione sono segnalati con (\*).

**Esercizio 14.1.** (\*) Un albero  $N$ -ario generico è un albero i cui nodi possono avere un numero arbitrariamente grande di rami uscenti. Si definisca una struttura dati adatta a rappresentare un albero  $N$ -ario. Per semplicità si consideri il caso in cui i nodi contengono, come dati utili, dei semplici numeri interi. Possiamo rappresentare questa struttura dati associando ad ogni nodo, invece di una coppia di puntatori a nodi, come nel caso degli alberi binari, una lista di puntatori a nodo. Tale lista è una lista concatenata semplice.

Si scriva una funzione che conta il numero di nodi di un albero  $N$ -ario.

**Esercizio 14.2** (TDE 20/02/2018). (\*) Si considerino le seguenti definizioni di liste:

```
1 typedef struct Nodo_s {
2     int v;
3     struct Nodo_s *next;
4 } Nodo;
5 typedef Nodo * Lista;
6
7 typedef struct NodoCompresso {
8     int v, n;
9     struct NodoCompresso *next;
10 } NodoC;
11 typedef NodoC * ListaC;
```

1. Si codifichi in C la seguente funzione: `ListaC comprimi(Lista l)`. La funzione riceve in input una lista di interi  $l$  e restituisce una nuova lista *compressa* secondo la seguente regola: per ogni elemento di  $l$  si memorizza nella nuova lista l'elemento stesso e il numero di ripetizioni consecutive in  $l$ . Ad esempio, se  $l=(3, 3, 3, 3, 2, 2, 3, 5, 5, 5)$ , allora la lista restituita sarà  $((3, 4), (2, 2), (3, 1), (5, 3))$ .
2. Si codifichi in C la seguente funzione: `ListaC comprimi2(Lista l)`. La funzione riceve in input una lista di interi  $l$  e restituisce una nuova lista costruita secondo la seguente regola: per ogni elemento di  $l$  si memorizza nella nuova lista l'elemento stesso e il numero totale di ripetizioni in  $l$  ordinate in base al valore dell'elemento stesso. Ad esempio se  $l=(3, 3, 3, 3, 2, 2, 3, 5, 5, 5)$  allora la lista restituita è  $((2, 2), (3, 5), (5, 3))$ .