

# Ricorsione e modelli di calcolo

6-7 ottobre 2025

## Esercizi sulla ricorsione.

**Esercizio 1.** Scrivete lo pseudocodice di una funzione **ricorsiva** che, **senza usare cicli**, ricevuti due numeri interi  $a, b$ , calcoli la somma di tutti i numeri nell'intervallo di interi  $[a, b]$ .

**Esercizio 2.** Scrivete lo pseudocodice di una funzione **ricorsiva** che, **senza usare cicli**, ricevuto in input un array di interi  $A[1, \dots, n]$  e un indice  $i \in [1, n]$ , restituisca il valore massimo in  $A[i, \dots, n]$ . Implementate la funzione in Go e verificatene la correttezza.

**Esercizio 3.** Scrivete lo pseudocodice di una funzione **ricorsiva** che, **senza usare cicli**, ricevuta in input una stringa  $s[1, \dots, n]$ , restituisce TRUE se  $s$  è un palindromo (cioè si legge uguale sia da destra che da sinistra), FALSE altrimenti. Suggerimento: i casi base sono due, a seconda che la lunghezza  $n$  della stringa sia pari o dispari! Implementate la funzione in Go e verificatene la correttezza.

**Esercizio 4.** Scrivete lo pseudocodice di una funzione **ricorsiva** che, **senza usare cicli**, ricevuta in input una stringa  $s[1, \dots, n]$ , un carattere  $c$  e un indice  $i \in [1, n]$ , restituisce il numero di occorrenze di  $c$  in  $s[i, \dots, n]$ . Implementate la funzione in Go e verificatene la correttezza.

**Esercizio 5.** Scrivete lo pseudocodice di una funzione **ricorsiva** che, **senza usare cicli**, ricevuto in input un array di interi  $A[1, \dots, n]$ , restituisce TRUE se  $A$  è ordinato in ordine crescente, FALSE altrimenti. Implementate la funzione in Go e verificatene la correttezza.

**Esercizio 6.** Scrivete lo pseudocodice di una funzione **ricorsiva** che, **senza usare cicli**, ricevuto in input un intero, restituisce il numero di zeri nella sua rappresentazione in base 10 (es: se l'input è 10350, la funzione restituisce il valore 2). Implementate la funzione in Go e verificatene la correttezza.

## Esercizi su modelli di costo e ricerca binaria.

**Esercizio 7.** Scrivete un programma che legge da stdin un intero  $n$  e restituisce  $n^n$ . Cosa succede per  $n$  abbastanza grande ( $n \geq 16$ )? Modificate il codice usando il package `math/big` per rendere il programma corretto anche per  $n \geq 16$ .

**Esercizio 8.** Scrivete un programma che legge da stdin un intero  $n$  e un secondo intero  $q$ , genera in maniera casuale una slice ordinata di interi di lunghezza  $n$ , e cerca  $q$  nella slice utilizzando la versione iterativa della ricerca binaria, stampando l'indice di una posizione in cui si trova  $q$  o “ $q$  non trovato” altrimenti. Scrivete esplicitamente la funzione che esegue la ricerca binaria.

**Esercizio 9.** Scrivete un programma che esegue lo stesso compito dell'esercizio 8, ma genera una slice di interi che abbiano tutti almeno 50 cifre. Interi di questo tipo non si possono memorizzare come `int`: usate il package `math/big`. Confrontate il tempo di esecuzione dei due programmi a parità di lunghezza  $n$  della slice generata, al crescere di  $n$  ( $n = 10^5, n = 10^6, n = 10^7, n = 10^8$ ). Cosa se ne può dedurre?