

## Corso di Algoritmi e Strutture Dati—Modulo 2

Esercizi su Divide-et-Impera – 31 Marzo 2023

Moreno Marzolla, Jocelyne Elias

**Esercizio 1.** Consideriamo un array  $A[1..n]$  composto da  $n \geq 0$  valori reali, non necessariamente distinti. L'array è ordinato in senso non decrescente. Scrivere un algoritmo ricorsivo di tipo divide-et-impera che restituisca *true* se e solo se  $A$  contiene valori duplicati. Calcolare il costo computazionale dell'algoritmo proposto.

**Esercizio 2.** Scrivere un algoritmo ricorsivo di tipo divide-et-impera che, dato un array  $A[1..n]$  di valori reali, restituisce *true* se e solo se  $A$  è ordinato in senso non decrescente, cioè se  $A[1] \leq A[2] \leq \dots \leq A[n]$ . Calcolare il costo computazionale dell'algoritmo proposto.

**Esercizio 3.** Si consideri un array  $A[1..n]$  composto da  $n \geq 1$  interi distinti ordinati in senso crescente ( $A[1] < A[2] < \dots < A[n]$ ). Scrivere un algoritmo efficiente che, dato in input l'array  $A$ , determina un indice  $i$ , se esiste, tale che  $A[i] = i$ . Nel caso esistano più indici che soddisfano la relazione precedente, è sufficiente restituirne uno qualsiasi. Determinare il costo computazionale dell'algoritmo.

**Esercizio 4.** Consideriamo un insieme di  $n$  variabili  $x_1, \dots, x_n$ . Sono dati un insieme di vincoli di uguaglianza della forma “ $x_i = x_j$ ”, e un altro insieme di vincoli di disuguaglianza della forma “ $x_i \neq x_j$ ”. Il problema consiste nel capire se tutti i vincoli possono essere soddisfatti. Ad esempio, considerando quattro variabili  $x_1, x_2, x_3, x_4$  soggette ai vincoli seguenti:

$$\begin{aligned}x_1 &= x_2; \\x_2 &= x_3; \\x_3 &= x_4; \\x_1 &\neq x_4\end{aligned}$$

risulta che in questo caso i vincoli non sono soddisfacibili. Descrivere a parole un algoritmo efficiente che, dati in input il numero  $n$  di variabili e le liste dei vincoli di uguaglianza e disuguaglianza, restituisce *true* se e solo se i vincoli sono soddisfacibili.

**Esercizio 5.** Si consideri un array  $A[1..n]$  contenente valori reali *ordinati in senso non decrescente/crescente*; l'array può contenere valori duplicati. Scrivere un algoritmo ricorsivo di tipo divide-et-impera che, dato  $A$  e due valori reali *low* ed *up* ( $low < up$ ), calcola quanti valori di  $A$  appartengono all'intervallo  $[low, up]$ . Determinare il costo computazionale dell'algoritmo proposto.