

# INTRODUZIONE AGLI ALGORITMI

## Esame Scritto a canali unificati

docenti: T. CALAMONERI, A. MONTI  
Sapienza Università di Roma  
31 Gennaio 2023

### Esercizio 1 (10 punti):

Si consideri il seguente algoritmo ricorsivo che, dato un array  $A$  di dimensione  $n$ , verifica se esistono due indici diversi  $i$  e  $j$  compresi nell'intervallo  $[0, n - 1]$  tali che  $A[i] = j$  e  $A[j] = i$ :

```
def IndiciValori(A, sx, dx):  
    if (sx >= dx): return False  
    else:  
        trovato = False  
        centro = (sx + dx) // 2  
        for i in range(sx, centro + 1):  
            for j in range(centro + 1, dx + 1):  
                if (A[i] == j) and (A[j] == i): trovato = True  
        trovato1 = IndiciValori(A, sx, centro)  
        trovato2 = IndiciValori(A, centro + 1, dx)  
        return trovato or trovato1 or trovato2
```

- Si imposti la relazione di ricorrenza che definisce il tempo di esecuzione giustificando dettagliatamente l'equazione ottenuta.
- Si risolva la ricorrenza usando **due metodi** a scelta, dettagliando i passaggi del calcolo e giustificando ogni affermazione.

**Esercizio 2 (10 punti):** Scrivere un algoritmo `ElementoPiuFrequente` che, dato un array  $A$  di  $n$  interi, compresi tra 1 e  $10n$  restituisce il valore più presente all'interno dell'array, a parità di occorrenze va restituito il valore minimo.

Ad esempio, se  $A = [2, 6, 8, 5, 2, 3, 6, 8, 9, 5, 8, 1, 2]$ , allora la risposta è 2 in quanto 2 ed 8 sono gli unici valori che compaiono 3 volte all'interno dell'array, mentre gli altri valori compaiono al più 2 volte.

Il costo computazionale dell'algoritmo proposto deve essere  $\Theta(n)$ .

Dell'algoritmo proposto:

- a) si scriva lo pseudocodice opportunamente commentato,
- b) si giustifichi il costo computazionale.

**Esercizio 3 (10 punti):** Sia  $L$  una lista concatenata semplicemente puntata data tramite il puntatore  $p$  alla sua testa e contenenti chiavi intere positive. Ogni record è composto da due campi: il campo *key* che contiene il valore del nodo ed il campo *next* che contiene il puntatore al nodo successivo della lista se questo esiste, il valore None altrimenti. Si progetti un **algoritmo ricorsivo** con costo computazionale  $O(n)$  che restituisca un puntatore al primo elemento della lista la cui chiave sia esattamente uguale alla somma delle chiavi di tutti gli elementi precedenti; se un tale elemento non esiste, verrà ritornato None.

Ad esempio, per la lista  $p \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 6$  verrà restituito un puntatore al record contenente l'informazione 3; si noti che anche il record contenente l'informazione 6 soddisfa la richiesta di avere la chiave pari alla somma dei precedenti, ma il record contenente 3 lo precede.

Dell'algoritmo proposto:

- a) si scriva lo pseudocodice opportunamente commentato,
- b) si giustifichi il costo computazionale trovando e risolvendo l'equazione di ricorrenza.