

# INTRODUZIONE AGLI ALGORITMI

## Esame Scritto a canali unificati

docenti: T. CALAMONERI, A. MONTI  
Sapienza Università di Roma  
Ottobre 2022

### Esercizio 1 (10 punti):

Si consideri la seguente funzione:

```
def es1(n):  
    if n <= 1: return 5  
    a = es1(n//2)  
    i = j = 1  
    while j < n:  
        j* = 2  
        i+ = 1  
    u, j = 1, n  
    while j > 1:  
        j- = i  
        u+ = 1  
    return a + es1(n//2) + u
```

- a) Si imposti la relazione di ricorrenza che ne definisce il tempo di esecuzione giustificando dettagliatamente l'equazione ottenuta.
- b) Qualora sia possibile, risolvere la ricorrenza utilizzando il **teorema principale** dettagliando il caso del teorema ed i passaggi logici. Se il teorema principale non è applicabile spiegarne il motivo.

### Esercizio 2 (10 punti):

Sia  $A$  un array di  $n$  elementi con  $n$  pari, contenente uno stesso numero di interi pari ed interi dispari. Un *riarrangiamento* degli interi in  $A$  è *valido* se nelle posizioni ad

indice pari compaiono interi pari e in quelle ad indice dispari interi dispari (l'indice 0 è considerato pari).

Ad esempio, per  $A = [7, 3, 1, 8, 8, 2, 1, 4]$  esistono diversi riarrangiamenti validi come:  $[8, 7, 2, 3, 8, 1, 4, 1]$ , oppure  $[4, 1, 2, 1, 8, 3, 8, 7]$  o anche  $[2, 3, 8, 1, 8, 7, 4, 1]$ .

Progettare un algoritmo che, preso l'array  $A$ , produca un riarrangiamento valido. L'algoritmo deve avere costo computazionale  $O(n)$  e deve utilizzare uno spazio di lavoro costante (in altri termini, non è possibile utilizzare liste concatenate o array di appoggio).

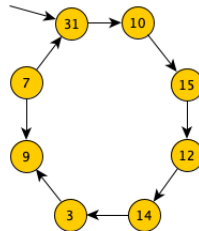
Dell'algoritmo proposto:

- a) si dia la descrizione a parole,
- b) si scriva lo pseudocodice,
- c) si giustifichi il costo computazionale.

### Esercizio 3 (10 punti):

Si consideri una lista a puntatori circolare  $L$  data tramite un puntatore  $p$  ad un suo elemento. In  $L$  ogni nodo ha 2 campi: il campo **key** contenente un intero ed il campo **next** con il puntatore al nodo seguente.

Sappiamo che gli interi dei vari nodi sono tutti distinti e bisogna trovare il valore minimo tra questi. Ad esempio per la lista circolare di seguito il valore cercato è 3:



Progettare un algoritmo **iterativo** che, dato il puntatore  $p$  ad un nodo della lista circolare, restituisce il valore cercato in tempo  $\Theta(n)$  dove  $n$  è il numero di nodi della lista.

Dell'algoritmo proposto:

- a) si dia la descrizione a parole,
- b) si scriva lo pseudocodice,
- c) si giustifichi il costo computazionale.