Introduzione agli Algoritmi
Esame Scritto a canali unificati

docenti: T. Calamoneri, A. Monti
Sapienza Università di Roma
31 Marzo 2022

Esercizio 1 (10 punti):
Si consideri la seguente funzione:

```
funzione Exam(n):
    tot ← 1;
    if n <= 1:  return tot;
    j ← 63;
    while j > 0 do:
        k ← 0;
        while 3 * k <= n do:  k ← k + 1;
        tot ← tot + 2 * Exam(k);
        j ← j − 7;
    while k > 0 do:
        for i = 1 to n do:  tot ← tot − 1
        k ← k − 1;
    return tot
```

a) Si imposti la relazione di ricorrenza che ne definisce il tempo di esecuzione giustificando dettagliatamente l'equazione ottenuta.

b) Si risolva la ricorrenza usando il **metodo dell'albero** dettagliando i passaggi del calcolo e giustificando ogni affermazione.

NOTA: Se necessario, usare le seguenti convenzioni:
- anziché $\leq$ o $\geq$ scrivere <= o >=
- anziché Θ e Ω scrivere Teta e Omega
- anziché $\sum_{i=0}^{k}$ scrivere S[i=0, k]
- anziché a^b scrivere a**b.

Esercizio 2 (10 punti): Dato un array **ordinato** A di n interi ed un intero k vogliamo sapere quante coppie in A hanno somma k. Si progetti un algoritmo **iterativo** che risolva il problema in tempo $\Theta(n)$.

Ad esempio:

- se $A = [1, 2, 2, 3, 4, 5, 5, 5, 8, 9, 9]$ e $k = 7$ l'algoritmo deve restituire 7 (le coppie a somma 7 sono infatti $(1, 5)$, $(1, 6)$, $(1, 7)$, $(2, 5)$, $(2, 6)$, $(2, 7)$ e $(3, 4)$).
- se $A = [1, 5, 5, 5, 9]$ e $k = 10$ l'algoritmo deve restituire 4 (le coppie a somma 10 sono infatti $(0, 4), (1, 2), (1, 3), (2, 3)$).

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 L, in cui ogni elemento è un record a tre campi: il campo `val` contenente un bit (cioé un valore 0 o 1), il campo `next` con il puntatore al nodo seguente (`next` vale *None* per l'ultimo record della lista) ed il campo `prec` con il puntatore al nodo precedente (`prec` vale *None* per il primo record della lista).

Bisogna verificare se la stringa che si ottiene considerando i bit dei vari nodi della lista è palindroma. Ad esempio, se la lista L in input è quella di sinistra nella figura che segue, la risposta è NO (la stringa binaria 010110 non è palindroma, mentre se L è la lista di destra la risposta è SI (la stringa binaria 11011 è palindroma).

Progettare un algoritmo (iterativo o ricorsivo) che, dato il puntatore s alla testa della lista, risolve il problema in tempo $\Theta(n)$, dove n è il numero di nodi della lista a puntatori. Lo spazio di lavoro dell'algoritmo proposto deve essere $O(1)$ (in altri termini NON è possibile definire e utilizzare altre liste).
Dell'algoritmo proposto:

a) si dia la descrizione a parole,

b) si scriva lo pseudocodice,

c) si giustifichi il costo computazionale.