

**ANNO ACCADEMICO 2019/2020 - Algoritmi e Strutture Dati - 25 febbraio 2021**

n. quesito	1	2	3	4	tot
punti	8	8	8	9	33

**Quesito 1**

Calcolare la complessità in funzione di  $n$  dell'istruzione:  $y = g(f(n))$ ; e il valore finale di  $y$ .  
con le seguenti definizioni di funzione. Indicare le eventuali relazioni di ricorrenza e spiegare brevemente il calcolo della complessità dei cicli.

```
int f(int x) {
    if (x<=1) return 1;
    int a=0; int b=0;
    for (int i=1; i<= g(x); i++)
        a+=i;
    return f(x-1)+a;
}

int g(int x) {
    if (x<=1) return 10;
    int b=0;
    for (int i=1; i<=x; i++)
        b+=i;
    return b+2*g(x/2);
}
```

**Funzione g**

numero iterazioni del for:  $O(n)$   
complessità di una iterazione:  $O(1)$   
tempo del for:  $O(n)$

tempo di g  
 $T(1) = d$   
 $T(m) = c \cdot n + T(n/2) \quad O(n)$

Risultato di g  
 $R(1) = d$   
 $R(m) = c \cdot n^2 + 2T(n/2) \quad O(n^2)$

**Funzione f**

numero iterazioni del for:  $O(n^2)$   
complessità di una iterazione:  $O(n)$   
tempo del for:  $O(n^3)$

tempo di f  
 $T(1) = d$   
 $T(m) = c \cdot n^3 + T(n-1) \quad O(n^4)$

Risultato di f  
 $R(1) = d$   
 $R(m) = c \cdot n^4 + T(n-1) \quad O(n^5)$

Tempo per calcolare  $g(f(n))$  :  $C[f(n)] + C[g(n^5)] = O(n^4) + O(n^5) = O(n^5)$

Valore di  $y$ :  $g(f(n)) = g(n^5) = O(n^{10})$

## quesito 2

- a) 2 Dare la definizione di **albero binario bilanciato e quasi bilanciato** e fare esempi.
- b) 2 Dare la definizione di **albero pienamente binario** e fare un esempio.
- c) 4 Scrivere una funzione che, dato un **albero generico t** con etichette intere memorizzato figlio/fratello, somma ad ogni nodo foglia l'etichetta del padre.

```
void somma(Node* t, int e=0){
    if (!t)
        somma(t->left, t->label);
    somma(t->right, e);
    if (!t->left) t->label += e;
}
```

## Quesito 3

- a) 2 A cosa serve l'algoritmo PLSC? A quale metodologia fa riferimento?
- b) 2 Qual è la sua complessità e spiegare come si calcola.
- c) 4 Applicarlo alle due sequenze "ABCEAD" e "CBACAED" e indicare tutte le PLSC .  
Mostrare tutti i passaggi in una tabella fatta come nel disegno.

		A	B	C	E	A	D
	0	0	0	0	0	0	0
C	0	0	0	1	1	1	1
B	0	0	1	1	1	1	1
A	0	1	1	1	1	2	2
C	0	1	1	2	2	2	2
A	0	1	1	2	2	3	3
E	0	1	1	2	3	3	3
D	0	1	1	2	3	3	4

**PLSC:** BCAD, ACED, BCED, ACAD

## Quesito 4

- a) 3 Definire gli **alberi di decisione** e per cosa vengono utilizzati. Applicarli per trovare il limite inferiore alla complessità degli algoritmi di **ordinamento** basati su confronti. Dire quali degli algoritmi di ordinamento basati su confronti visti a lezione sono ottimi e quali no.
- b) 3 Descrivere l'algoritmo di ordinamento **quicksort** e indicare la sua complessità nel caso medio e nel caso peggiore. Indicare un array che rappresenta un "caso peggiore".
- c) 3 Fare un esempio di programma dove una **eccezione** viene lanciata all'interno di un blocco try appartenente a una funzione ma viene gestita (correttamente) dal main.