ANNO ACCADEMICO 2019/2020 - Algoritmi e Strutture Dati - 28 luglio 2020
TURNO B

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

quesito 1

Calcolare la complessità del seguente comando in funzione del numero di nodi dell'albero binario quasi bilanciato t:

```
for (int i=1; i <=f(g(n)); i++) a++;
```

Indicare per esteso le relazioni di ricorrenza e, per ogni comando ripetitivo, il numero di iterazioni e la complessità della singola iterazione.

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

Funzione f

$T_f(0)= d$
$T_f(n)= cn + T_f(n-1)$ $T_f(n)$ è $O(n^2)$

$R_f(0)= 1$
$R_f(n)= n^2+ R_f(n-1)$ $R_f(n)$ è $O(n^3)$

Funzione g

$T_g(0,1)=d$
$T_g(n)= cn^3+ T_g(n/2)$ $T_g(n)$ è $O(n^3)$

$R_g(0)=d$
$R_g(n)= n^3 + 2R_g(n/2)$ $R_g(n)$ è $O(n^3)$

Calcolo del comando:
numero iterazioni: $R_f(R_g(n)) = R_f(n^3) = O(n^9)$
complessità della singola iterazione: $T_g(n) + T_f(n^3) = O(n^3) + O(n^6)= O(n^6)$
complessità del for: $O(n^{15})$

quesito 2

Scrivere una funzione `int conta(Node* t)`, che, dato un albero generico `t` memorizzato figlio/fratello, conta i nodi per cui il primo figlo ha lo stesso numero di figli dell'ultimo figlio. Se un nodo non ha figli non viene contato. Se ha un solo figlio, questo è anche l'ultimo.

```
int conta (Node* t) {
   if (!t) return 0;
   if (!t->left) return conta (t->right);
   return conta (t->left) + conta(t->right) +
           (contafigli(t->left)==contafigli(ultimo(t->left)->left));
}

int contafigli (Node* t) {
   if (!t) return 0;
   return 1+ contafigli (t->right);
}

Node * ultimo (Node* t) {
   if (!t->right) return t;
   return ultimo (t->right);
}
```

quesito 3: PLSC

a) **1** A cosa serve l'algoritmo PLSC? A quale metodologia fa riferimento?
b) **2** Qual è la sua complessità e come si calcola?
c) **2** Indicare come potrebbe essere progettato un algoritmo che si basa sulla stessa metodologia per trovare l'ennesimo numero della successione di Fibonacci. **Riempire un array di n posizioni con i numeri di Fibonacci tramite un comando ripetitivo.**
d) **3** Applicarlo alle due sequenze "ABCEAD" e "CABACADED" 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
A	0	1	1	1	1	2	2
B	0	1	2	2	2	2	2
A	0	1	2	2	2	3	3
C	0	1	2	3	3	3	3
A	0	1	2	3	3	4	4
D	0	1	2	3	3	4	5
E	0	1	2	3	4	4	5
D	0	1	2	3	4	4	5

PLSC: ABCED, ABCAD

Quesito 4

a) **3** Dare la definizione di complessità computazionale. Confrontare le funzioni seguenti dal punto di vista della complessità: dire se una è O dell'altra indicando una coppia (n0,c) e, in caso contrario, spiegare perchè non lo è.

$$F(n) = \begin{cases} 20\ n^3 + n & n \text{ divisibile per } 10 \\ 100\ n^3 & n \text{ non divisibile per } 10 \end{cases}$$

$$G(n) = \begin{cases} 90\ n^3 & n \text{ pari} \\ n^3 & n \text{ dispari} \end{cases}$$

F è O(g) : [n0= 1 , c= 100]
G è O(f) : [n0= 1 , c= 5]

b) **3** Spiegare la differenza fra **albero binario** e **albero generico**. Quali **visite** vengono usate per questi alberi e come sono definite? Quali sono le **memorizzazioni** viste a lezione? Se l'albero generico è memorizzato figlio/fratello, qual è la **relazione** fra le sue visite e quelle del trasfomato binario?

c) **3** Quali sono le regole per gli specificatori di accesso **private, public e protected** (assumere la derivazione pubblica)? Quale di questi specificatori non ha significato se non si ha ereditarietà fra classi?