

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

quesito 1

Calcolare la complessità del seguente comando in funzione di n:

```
for (int i=0; i <= g(n); i++) cout << f(n);
```

con le funzioni **f** e **g** definite come segue. 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 1;
    cout << f(x/3);
    return 1+ f(x/3) + 8* f(x/3);
}
```

```
int g(int x) {
    int a=0;
    for (int i=0; i <= f(x); i++) a++;
    for (i=0; i <= f(x)/x; i++) a+=i;
    return a;
}
```

$T_f(0,1)=d$

$T_f(n)=c + 3 T_f(n/3)$ $T_f(n)$ è $O(n)$

$R_f(0,1)=1$

$R_f(n)=1 + 9 R_f(n/3)$ $R_f(n)$ è $O(n^2)$

Calcolo $T_g(n)$

1 for:

numero iterazioni: $O(n^2)$ Complessità singola iterazione: $O(n)$

complessità dei for: $O(n^3)$

2for:

numero iterazioni: $O(n)$ Complessità singola iterazione: $O(n)$

complessità dei for: $O(n^2)$

il valore di a alla fine è' $O(n^2)$

$T_g(n)$ è $O(n^3) + O(n^2)= O(n^3)$

$R_g(n)$ è $O(n^2)$

Calcolo for del blocco:

numero iterazioni: $R_g(n)= O(n^2)$

Complessità singola iterazione: $T_g(n)+ T_f(n)+= O(n^3) + O(n)=O(n^3)$

Complessità totale blocco = $O(n^5)$

quesito 2

- a) 2 Dare la definizione di albero generico.
- b) 2 Definire le visite viste a lezione su di un albero generico.
- c) 4 Scrivere una funzione **void aggiungi(Node* t, int x)** che, dato un albero generico **t** con etichette intere diverse fra di loro memorizzato figlio/fratello e un intero **x**, aggiunge al nodo con etichetta **x**, se esiste, un primo figlio avente come etichetta il numero di nodi del sottoalbero di radice **x**.

```
void aggiungi (Node* t, int x) {  
    if (!t) return;  
    if (t->label ==x) {  
        Node * n= new Node;  
        n->label= Nodes(t);  
        n->left=NULL;  
        n->right=t->left;  
        t->left=n;  
        return;  
    }  
    aggiungi (t->left, x);  
    aggiungi (t->right, x);  
}
```

quesito 3

- a) 2 Definire il tipo di dato max-heap, la sua memorizzazione, le sue operazioni con la relativa complessità.
- b) 2 Indicare per la implementazione di quali algoritmi visti a lezione viene utilizzato uno heap (max-heap o min-heap) e come. Spiegare.
- c) 4 Dato lo heap:

[100, 90, 96, 80, 75, 91, 90, 70]

indicare lo stato dello stesso dopo le due successive operazioni:

- i. inserimento di 95
- ii. estrazione

indicando anche le chiamate a up e down.

Heap iniziale	100, 90, 96, 80, 75, 91, 90, 70	Chiamate a up e down
Dopo l'inserimento di 95	100, 95, 96, 90, 75, 91, 90, 70, 80	up(8), up(3), up(1)
Dopo una estrazione	96, 95, 91, 90, 75, 80, 90, 70	down(0), down(2), down (5)

quesito 4

- a) 3 Definire le classi P e NP, illustrare il teorema di Cook, definire i problemi NP-completi.
- b) 3 Definire il tipo di dato **albero binario di ricerca** con le sue operazioni e le relative complessità nel caso medio e nel caso peggiore. Fare un esempio di due alberi binari di ricerca diversi con gli stessi nodi.
- a) 3 Fare un esempio in c++ di gerarchia con due classi con una funzione **fun** nella classe base e un **main** che mostra un diverso comportamento se la funzione **fun** è **virtual** oppure no. Spiegare.