

Cognome e Nome:
Numero di Matricola:

Spazio riservato alla correzione

1	2	3	4	5	6	7	totale
/5	/6	/4	/25	/15	/20	/25	/100

- Indicare quali delle seguenti affermazioni sono vere e quali sono false.
 - $n^3+n^2+4 = \Theta(n^3)$
 - $n! = \Omega(2^n)$
 - $\log n = \Theta(\log(n^2))$
 - $\sqrt{\log n} = \Omega(\log \log n)$
 - $4^{\log_4 n} = \Theta(n)$
- Si dimostri in modo formale e facendo uso **direttamente** della definizione di O (**senza ricorrere ad altre proprietà**) che se $f(n) = O(p(n))$ e $g(n) = O(q(n))$ allora $f(n)g(n) = O(p(n)q(n))$.

3. Si analizzi il tempo di esecuzione del seguente segmento di codice fornendo una stima asintotica in termini di Θ per essi. **Si giustifichi in modo chiaro la risposta.**

```
i=0;
while(i<n){
  j=0;
  while(j<i){

    j=j+1;

  }
  i=i+1;
}
```

4.

a. Si scriva lo pseudocodice dell'algoritmo **QuickSort** che ordina in modo non decrescente un array di numeri (**per l'algoritmo Distribuzione, si veda il punto successivo**)

b. Si descriva in modo preciso e schematico il comportamento dell'algoritmo **Distribuzione** specificando anche l'input dell'algoritmo. Non è sufficiente fornire lo pseudocodice.

- c. Si analizzi il tempo di esecuzione dell'algoritmo **Distribuzione** nel **caso pessimo**, giustificando in modo chiaro la risposta.

- d. Si scriva la relazione di ricorrenza che esprime il tempo di esecuzione **nel caso pessimo** dell'algoritmo **QuickSort spiegando in modo chiaro come si ottiene la suddetta relazione.**
- e. Utilizzando la relazione di ricorrenza di cui al punto precedente, si fornisca una stima asintotica del tempo di esecuzione **nel caso pessimo** dell'algoritmo **QuickSort spiegando in modo chiaro come viene ottenuta la suddetta stima asintotica a partire dalla relazione di ricorrenza.**

5. Si scriva lo pseudocodice di un algoritmo $A(u)$ che prende in input un nodo u di un albero binario e restituisce true se ogni nodo interno dell'albero radicato in u ha esattamente due figli e false altrimenti.

6

- a. Si disegni l'heap che si ottiene effettuando l'Enqueue dei nove elementi con chiavi 30 40 2 10 7 8 11 95 6, in questo ordine, in un heap inizialmente vuoto.

b. Si disegni l'heap che si ottiene effettuando la Deque nell'heap disegnato al punto a.

c. Si descrivano in modo chiaro e schematico i passi dell'algoritmo Enqueue della coda a priorit  implementata con Heap descrivendo anche il comportamento degli algoritmi invocati da Enqueue. Occorre anche specificare l'input di Enqueue e degli algoritmi da esso invocati.

- d. Si fornisca il tempo di esecuzione dell'algorithmo Enqueue nel caso pessimo giustificando in modo chiaro la risposta.

7.

- a. Si descriva in modo chiaro e schematico il comportamento dell'algoritmo
Inserisci dell'albero binario di ricerca specificando anche l'input dell'algoritmo.

- b. Si fornisca una stima asintotica del tempo di esecuzione **nel caso pessimo** dell'algoritmo. Inserisci giustificando in modo chiaro la risposta.
- c. Si scriva lo pseudocodice dell'algoritmo $F(u)$ che prende in input un nodo u di un albero binario di ricerca e restituisce l'elemento con chiave massima dell'albero binario di ricerca avente u come radice.

- d. In un albero binario di ricerca **AVL** inizialmente vuoto vengono inserite le sei chiavi 10 7 8 9 4 3, in questo ordine. Per ogni inserimento effettuato, si disegni l'albero AVL che risulta dopo l'inserimento. Se l'inserimento di un elemento causa lo sbilanciamento dell'albero allora per quell'inserimento occorre disegnare l'albero prima e dopo il ribilanciamento indicando il nodo critico e di quale tipo (tra SS, SD, DD, DS) e' lo sbilanciamento.