Università degli Studi di L'Aquila

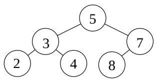
Prova intermedia dell'esame di **Algoritmi e Strutture Dati**

Martedì 5 Novembre 2002 – Prof. Guido Proietti

Scrivi i tuoi dati ⟹	Cognome:	Nome:	Matricola:	PUNTI
ESERCIZIO 1	Risposte Esatte:	Risposte Omesse:	Risposte Errate:	
ESERCIZIO 2				
ESERCIZIO 3	Correttezza:	Efficienza:	Analisi:	
TOTALE				

ESERCIZIO 1: Domande a risposta multipla (15 punti)

Premessa: Questa parte è costituita da 15 domande a risposta multipla. Per ciascuna domanda vengono fornite 4 risposte, di cui soltanto una è corretta. Per rispondere utilizzare la griglia annessa, barrando con una $\times$ la casella corrispondente alla risposta prescelta. È consentito omettere la risposta. In caso di errore, contornare con un cerchietto la $\times$ erroneamente apposta (ovvero, in questo modo $\otimes$) e rifare la $\times$ sulla nuova risposta prescelta. Se una domanda presenta più di una risposta, verrà considerata omessa. Per tutti i quesiti verrà attribuito un identico punteggio, e cioè: risposta esatta 3 punti, risposta omessa 0 punti, risposta sbagliata -1 punto. Il voto relativo a questa parte è ottenuto sommando i punti ottenuti e normalizzando su base 15. Se tale somma è negativa, verrà assegnato 0.

1. L'algoritmo Insertion Sort, nel caso medio costa:
   a) $O(n)$ b) $o(n^2)$ c) $\Omega(n^2)$ d) $\Theta(n \log n)$
2. L'algoritmo Insertion Sort, nel caso migliore costa:
   a) $\Omega(n \log n)$ b) $\omega(n)$ c) $\Theta(n)$ d) $\Theta(n \log n)$
3. Se $f(n) = \Omega(g(n))$ e $g(n) = \omega(h(n))$, allora:
   a) $h(n) = \Omega(f(n))$ b) $f(n) = O(h(n))$ c) $f(n) = \Theta(h(n))$ d) $f(n) = \omega(h(n))$
4. Quale delle seguenti relazioni asintotiche è falsa:
   a) $n \log^3 n = O(n \log n^3)$ b) $n = \Theta(2^{\log n})$ c) $2^{n+1} = \Theta(2^n)$ d) $n \log n^4 = \Theta(n \log n)$
5. L'algoritmo Selection Sort, nel caso migliore costa:
   a) $o(n^2)$ b) $\Theta(n^2)$ c) $O(n \log n)$ d) $\Theta(n \log n)$
6. L'algoritmo Bubble Sort, nel caso peggiore costa:
   a) $o(n^2)$ b) $\Omega(n^2)$ c) $\omega(n^2)$ d) $\Theta(n \log n)$
7. Il numero di foglie dell'albero di decisione associato al problema dell'ordinamento è:
   a) $\Theta(n \log n)$ b) $\omega(n!)$ c) $O(n \log n)$ d) $\Omega(n!)$
8. L'algoritmo Merge Sort, nel caso migliore costa:
   a) $O(n)$ b) $O(n^2)$ c) $\omega(n \log n)$ d) $\Theta(n)$
9. Per $n = 4^k$, la soluzione dell'equazione di ricorrenza $T(n) = 2 \cdot T(n/4) + n, T(1) = \Theta(1)$, è:
   a) $\Theta(n^2)$ b) $\Theta(n^{\log_4 2})$ c) $O(n)$ d) $\Theta(n \log n)$
10. Quale dei seguenti vettori non rappresenta un heap:
    a) A=[5,3,4,1,2] b) A=[20,19,12,13,14,15] c) A=[5,4,3,2,1] d) A=[5]
11. La procedura *Heapify*$(A, 1)$ applicata al vettore $A = [12, 15, 21, 14, 9, 13]$ restituisce:
    a) $A = [21, 15, 12, 14, 9, 13]$ b) $A = [15, 12, 21, 14, 9, 13]$ c) $A = [15, 14, 21, 12, 9, 13]$ d) $A = [21, 15, 13, 14, 9, 12]$
12. L'algoritmo *Build-Heap*(A) nel caso peggiore costa:
    a) $\Theta(n \log n)$ b) $\Omega(n \log n)$ c) $O(1)$ d) $O(n)$
13. La delimitazione inferiore al problema della ricerca in un insieme ordinato di n elementi è:
    a) $\Theta(n \log n)$ b) $\Theta(n)$ c) $\Omega(\log n)$ d) $\Omega(n \log n)$
14. Dato l'albero binario  la visita in ordine anticipato restituisce:

    a) 2,3,4,5,8,7 b) 5,3,2,4,7,8 c) 2,4,3,8,7,5 d) 8,7,5,4,3,2
15. Un albero binario di ricerca di altezza k contiene:
    a) meno di $2^{k+1} - 1$ elementi b) tra $k + 1$ e $2^{k+1} - 1$ elementi c) almeno 2^k elementi d) tra 2^k e $2^{k+1} - 1$ elementi

Griglia Risposte

	Domanda														
Risposta	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a															
b															
c															
d															

ESERCIZIO 2: Domande a risposta aperta (6 punti)

Premessa: Questa parte è costituita da 2 domande a risposta aperta. Rispondere ad **una sola** domanda selezionata a piacere. La risposta giudicata corretta ed esaustiva è valutata 6 punti.

1. Fornire la dimostrazione del teorema principale, con le ipotesi restrittive definite a lezione.
2. Descrivere l'algoritmo Heap Sort ed analizzarne la complessità computazionale in modo dettagliato.

ESERCIZIO 3: Realizzazione di un algoritmo (9 punti)

Premessa: L'esercizio verrà valutato soltanto se corredato da adeguata descrizione del funzionamento dell'algoritmo, ed in base ai seguenti parametri: correttezza algoritmo (4 punti), efficienza algoritmo (3 punti) ed analisi della complessità (2 punti).

Sia T un albero binario di ricerca con chiavi intere. Realizzare ed analizzare un algoritmo che, preso in input T ed un puntatore ad un elemento x di T, restituisca l'elemento con la più piccola chiave pari più grande della chiave di x. Si supponga di disporre di una funzione ausiliaria *Resto*$(N, 2)$, la quale restituisce il resto della divisione di un numero N per 2.