



# UNIVERSITÀ DEGLI STUDI DI L'AQUILA

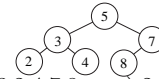
## Prova di Recupero di Algoritmi e Strutture Dati

Martedì 13 Luglio 2004 – 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 × la casella corrispondente alla risposta prescelta. È consentito omettere la risposta. In caso di errore, contornare con un cerchietto la × erroneamente apposta (ovvero, in questo modo ⊗) e rifare la × 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.

- L'algoritmo di ordinamento non crescente INSERTION SORT applicato ad  $A = [3, 4, 2, 1]$ , esegue un numero di confronti tra elementi pari a:  
a) 2   b) 3   c) 1   d) 4
- Quale delle seguenti funzioni  $f(n)$  è  $\Theta(n \log n)$ :  
a)  $f(n) = n \log n^3$    b)  $f(n) = n + \log n$    c)  $f(n) = n \log^2 n$    d)  $f(n) = n^2$
- L'algoritmo di ordinamento non decrescente SELECTION SORT applicato alla sequenza  $A = [5, 4, 3, 2, 1]$ , esegue un numero di confronti tra elementi pari a:  
a) 5   b) 4   c) 15   d) 10
- 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)  $\Omega(n!)$
- A quale delle seguenti classi non appartiene la complessità dell'algoritmo MERGE SORT:  
a)  $o(n \log n)$    b)  $\omega(n)$    c)  $O(n \log n)$    d)  $o(n^2)$
- Il numero massimo di heap binari distinti generati dall'insieme di 4 elementi  $A = \{1, 2, 3, 4\}$  è:  
a) 1   b) 4   c) 2   d) 3
- Dato un array ordinato in modo non decrescente, la procedura *Build-Heap* per la costruzione di un heap costa:  
a)  $\Theta(n \log n)$    b)  $\Theta(1)$    c)  $\Theta(n)$    d)  $\Theta(n^2)$
- Quale delle seguenti classi caratterizza meglio il costo dell'estrazione del massimo in una coda di priorità di  $n$  elementi realizzata con una lista lineare ordinata?  
a)  $O(n)$    b)  $\Theta(n)$    c)  $O(1)$    d)  $\Omega(n)$
- La delimitazione inferiore al problema della ricerca di un elemento in un insieme non ordinato di  $n$  elementi è:  
a)  $\Theta(\log n)$    b)  $\Theta(n \log n)$    c)  $\Omega(n)$    d)  $\Omega(n \log n)$
- Dato l'albero binario  la visita in ordine simmetrico 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
- Un grafo non connesso di 4 vertici, può avere al più un numero di archi pari a:  
a) 0   b) 3   c) 2   d) 4
- La visita in ampiezza dell'albero di cui alla Domanda 10 eseguita partendo dal nodo 3 restituisce un BFS di altezza:  
a) 2   b) 3   c) 1   d) 4
- Dato un grafo pesato con  $n$  vertici ed  $m$  archi rappresentato tramite liste di adiacenza, l'algoritmo di Dijkstra realizzato con un heap binario costa:  
a)  $\Theta(n^2)$    b)  $\Theta(n + m)$    c)  $O(n^2)$    d)  $O(m \log n)$
- Il problema della gestione di  $n$  insiemi disgiunti sottoposti ad  $m$  operazioni senza utilizzare l'euristica dell'unione pesata, può essere risolto in:  
a)  $\Theta(m + n \log n)$    b)  $\Theta(m)$    c)  $O(m^2)$    d)  $O(m + n \log n)$
- Dato un grafo pesato con  $n$  vertici ed  $m$  archi, l'algoritmo di Kruskal esegue un numero di operazioni FIND( $u$ ) pari a:  
a)  $O(m)$    b)  $\Theta(n)$    c)  $\Theta(m \log n)$    d)  $\Theta(m)$

### 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.

- Descrivere l'algoritmo MERGE SORT ed analizzarne la complessità computazionale in modo dettagliato.
- Descrivere dettagliatamente il problema della determinazione del *minimo albero ricoprente*, ed illustrare l'algoritmo di Prim.

### 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  $A$  un heap binario. Realizzare ed analizzare un algoritmo che, preso in input  $A$  ed un elemento  $x$  di  $A$ , restituisce l'elemento immediatamente più grande di  $x$  e quello immediatamente più piccolo.