Università Degli Studi Di L'Aquila

Prova di Recupero di **Algoritmi e Strutture Dati**

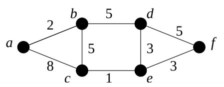
Martedì 6 Settembre 2005 – Prof. Guido Proietti

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 migliore costa:
   *a) $O(n)$ b) $\omega(n)$ c) $\Theta(n^2)$ d) $\Omega(n \log n)$
2. 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) $f(n) = \omega(h(n))$
3. L'algoritmo Selection Sort, nel caso medio costa:
   a) $o(n^2)$ *b) $\Theta(n^2)$ c) $\Theta(n \log n)$ d) $O(n \log n)$
4. L'algoritmo ottimale di fusione di due sequenze ordinate di lunghezza n e n^2 rispettivamente, ha complessità:
   a) $\Theta(n)$ b) $O(n)$ c) $\omega(n^2)$ *d) $\Theta(n^2)$
5. A quale delle seguenti classi non appartiene la complessità dell'algoritmo Merge Sort:
   a) *$o(n \log n)$ b) $\Omega(n)$ c) $O(n^2)$ d) $\Theta(n \log n)$
6. La soluzione dell'equazione di ricorrenza $T(n) = a \cdot T(n/b) + \Theta(n)$, con $T(1) = \Theta(1)$, a, b costanti non negative, $n = b^k$ ed $a > b$ è:
   a) $O(n)$ *b) $O\left(n^{\log_b a}\right)$ c) $O\left(n^{\log_a b}\right)$ d) $O(n \log n)$
7. Quale dei seguenti vettori rappresenta un heap binario:
   a) A=[10,9,6,7,5,11] *b) A=[20,16,9,15,12,8] c) A=[20,16,9,15,17,5,4] d) A=[5,3,2,1,6]
8. Dato un array non ordinato, la procedura Build-Heap per la costruzione di un heap costa:
   a) $\Theta(n \log n)$ b) $\Theta(1)$ *c) $O(n)$ d) $\Theta(n^2)$
9. Una coda di priorità realizzata con un heap binario supporta l'inserimento di un elemento in:
   *a) $O(\log n)$ b) $\Omega(n)$ c) $\Theta(\log n)$ d) $O(1)$
10. La delimitazione inferiore al problema della ricerca in un insieme di n elementi è:
    a) $\Theta(n \log n)$ *b) $\Omega(n)$ c) $O(\log n)$ d) $\Omega(n \log n)$
11. In un albero binario di ricerca di altezza h, il successore di un elemento può essere determinato in:
    a) $\Theta(\log h)$ b) $O(\log h)$ c) $\Theta(1)$ *d) $O(h)$
12. La visita in profondità del grafo  eseguita partendo dal nodo a non può restituire la sequenza di nodi:
    a) $acbdef$ *b) $abcdef$ c) $abdfec$ d) $acefdb$
13. Il peso del *massimo albero ricoprente* del grafo di domanda 12 è pari a:
    a) 24 *b) 26 c) 25 d) 14
14. Il nodo a distanza massima da c nel grafo di domanda 12 è:
    a) f *b) a c) d d) b
15. L'unione di 2 insiemi disgiunti A, B senza l'euristica dell'unione pesata costa nel caso peggiore:
    a) $\Theta(\min(|A|, |B|))$ *b) $\Theta(\max(|A|, |B|))$ c) $\Theta(|A| + |B|)$ d) $\Theta(|A| \cdot |B|)$

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 verrà valutata 6 punti.

1. Illustrare dettagliatamente la tecnica per la determinazione del lower bound del problema dell'*ordinamento* tramite confronti.
2. Descrivere i vari algoritmi per le operazioni di ricerca, minimo, massimo, predecessore e successore in un albero binario di ricerca, valutandone la complessità computazionale.

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).

Dato un grafo $G = (V, E)$ non orientato, pesato con pesi interi positivi e rappresentato mediante liste di adiacenza, e dato un intero k, realizzare un algoritmo che restituisca tutte le coppie di nodi di G che sono a distanza esattamente k l'uno dall'altro.