



UNIVERSITÀ DEGLI STUDI DI L'AQUILA

Prova di recupero di **Algoritmi e Strutture Dati**

Martedì 17 Luglio 2007 – Prof. Guido Proietti

Scrivi i tuoi dati =>

Cognome: Nome: Matricola:

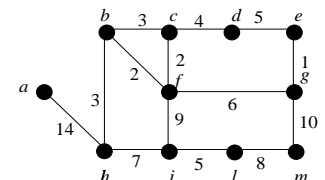
ESERCIZIO 1 (25 punti): Domande a risposta multipla

Premessa: Questa parte è costituita da 20 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 25. Se tale somma è negativa, verrà assegnato 0.

- Siano $f(n)$ e $g(n)$ i costi dell'algoritmo INSERTION SORT nel caso medio e dell'algoritmo SELECTION SORT in quello migliore, rispettivamente. Quale delle seguenti relazioni asintotiche è falsa:
 - *a) $f(n) = o(g(n))$ b) $f(n) = \Theta(g(n))$ c) $f(n) = O(g(n))$ d) $f(n) = \Omega(g(n))$
- Quale delle seguenti funzioni $f(n)$ è $\Theta(n)$:
 - a) $f(n) = n/\log n$ *b) $f(n) = n + \log n$ c) $f(n) = 1$ d) $f(n) = n^2$
- Un algoritmo ha una complessità temporale $O(f(n))$ se:
 - a) Il tempo di esecuzione $T(n)$ dell'algoritmo su uno specifico input di dimensione n verifica $T(n) = O(f(n))$
 - *b) Il tempo di esecuzione $T(n)$ dell'algoritmo su ogni input di dimensione n verifica $T(n) = O(f(n))$
 - c) Il tempo di esecuzione medio $T(n)$ dell'algoritmo su un input di dimensione n verifica $T(n) = O(f(n))$
 - d) Nel caso migliore, il tempo di esecuzione $T(n)$ dell'algoritmo su un input di dimensione n verifica $T(n) = O(f(n))$
- Un problema ha una delimitazione inferiore alla complessità temporale $\Omega(f(n))$ se:
 - *a) Tutti gli algoritmi per la sua risoluzione hanno una delimitazione inferiore alla complessità computazionale pari a $\Omega(f(n))$
 - b) Tutti gli algoritmi per la sua risoluzione hanno una delimitazione superiore alla complessità computazionale pari a $O(f(n))$
 - c) Esiste un algoritmo per la sua risoluzione che ha una delimitazione inferiore alla complessità computazionale pari a $\Omega(f(n))$
 - d) Esiste un algoritmo per la sua risoluzione che ha una delimitazione superiore alla complessità computazionale pari a $O(f(n))$
- Dato un problema con una delimitazione inferiore alla complessità temporale pari a $\Omega(f(n))$, un algoritmo per la sua risoluzione si dice *ottimale* se ha un tempo di esecuzione $g(n)$ tale che:
 - *a) $g(n) = \Theta(f(n))$ b) $g(n) = o(f(n))$ c) $g(n) = \omega(f(n))$ d) $g(n) = f(n)$
- La delimitazione superiore alla complessità temporale del problema della ricerca di un elemento in un insieme non ordinato è:
 - a) $\Theta(n^2)$ b) $O(\log n)$ c) $\omega(n)$ *d) $O(n)$
- Siano $f(n)$ e $g(n)$ i costi degli algoritmi HEAPSORT e QUICKSORT nel caso peggiore e in quello medio, rispettivamente. Quale delle seguenti relazioni asintotiche è vera:
 - a) $g(n) = o(f(n))$ *b) $f(n) = \Theta(g(n))$ c) $f(n) = \omega(g(n))$ d) $g(n) = \omega(f(n))$
- Qual è la complessità temporale dell'algoritmo INTEGER SORT applicato ad un array A di n elementi con valori in $[1..10^9]$?
 - *a) $\Theta(n)$ b) $\Theta(n^2)$ c) $O(10^9)$ d) $\Theta(n \log n)$
- Una coda di priorità realizzata con un heap binario supporta la ricerca del secondo elemento più grande in:
 - a) $O(\log n)$ *b) $O(n)$ c) $\Theta(\log n)$ d) $O(1)$
- Sia H_1 un heap binomiale di 5 elementi, e sia H_2 un heap binomiale di 3 elementi. Da quali alberi binomiali è formato l'heap binomiale ottenuto dalla fusione di H_1 e H_2 ?
 - *a) $\{B_3\}$ b) $\{B_0, B_1, B_2, B_2\}$ c) $\{B_4\}$ d) $\{B_8\}$
- Dato un albero binario di ricerca di n elementi, la cancellazione di un elemento restituisce un albero avente al massimo altezza:
 - *a) $n - 2$ b) n c) $\Theta(\log n)$ d) $n - 1$
- In un albero AVL di n elementi, l'inserimento di un elemento nel caso migliore induce un numero di rotazioni pari a:
 - *a) 0 b) 2 c) $\Theta(\log n)$ d) 1
- Sia $h(\cdot)$ una funzione hash. Quale delle seguenti funzioni descrive il metodo di scansione lineare in una tabella hash di dimensione m per l'inserimento di un elemento con chiave k dopo l' i -esima collisione:
 - a) $c(k, i) = (h(k) + m) \bmod i$ b) $c(k, i) = h(k) \bmod m$ *c) $c(k, i) = (h(k) + i) \bmod m$ d) $c(k, i) = i \bmod m$
- In un grafo *completo* e non pesato con n vertici, il cammino semplice di lunghezza massima tra due vertici fissati è lungo:
 - a) 1 b) 2 c) $n(n-1)/2$ *d) $n - 1$
- La visita in ampiezza del grafo  eseguita partendo dal nodo b genera un albero BFS di altezza pari a:
 - *a) 1 b) 2 c) 3 d) 4
- Dato il grafo di domanda (15), quanto pesa l'arco leggero del taglio $(\{a, b, e\}, \{c, d\})$?
 - *a) 3 b) 2 c) 4 d) non esiste
- Sia d_{xy}^k il costo di un cammino minimo k -vincolato da x a y , secondo la definizione di Floyd e Warshall. Risulta:
 - a) $d_{xy}^k = \min\{d_{xy}^{k-1}, d_{xv_k}^{k-1} + d_{v_kx}^{k-1}\}$ *b) $d_{xy}^k = \min\{d_{xy}^{k-1}, d_{xv_k}^{k-1} + d_{v_ky}^{k-1}\}$
 - c) $d_{xy}^k = \min\{d_{xy}^{k-1}, d_{xv_k}^k + d_{v_ky}^k\}$ d) $d_{xy}^k = \min\{d_{xy}^k, d_{xv_k}^{k-1} + d_{v_ky}^{k-1}\}$
- Il grado dell'*albero dei cammini minimi* radicato in c del grafo della domanda (15) è: a) 1 *b) 2 c) 3 d) 4
- Il *minimo albero ricoprente* del grafo della domanda (15) ha peso totale: a) 4 *b) 13 c) 14 d) 5
- Dato un grafo pesato con n vertici ed $m = O(n)$ archi, l'algoritmo di Prim realizzato con heap di Fibonacci costa:
 - a) $\Theta(n^2)$ b) $\Theta(n + m)$ c) $O(m)$ *d) $O(n \log n)$

Griglia Risposte

	Domanda																			
Risposta	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a																				
b																				
c																				
d																				



ESERCIZIO 2 (5 punti) (Da svolgere sul retro della pagina!)

Mostrare l'intera esecuzione, passo per passo, dell'algoritmo di Kruskal sul seguente grafo.