



UNIVERSITÀ DEGLI STUDI DI L'AQUILA

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

Martedì 18 Settembre 2007 – Prof. Guido Proietti

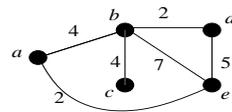
Scrivi i tuoi dati \Rightarrow

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.

- L'algoritmo di ordinamento crescente INSERTION SORT applicato ad una sequenza di input ordinata in modo decrescente esegue un numero di confronti tra elementi pari a:
a) $n - 1$ b) $n \log n$ c) $n + 1$ *d) $n(n - 1)/2$
- Siano $f(n)$ e $g(n)$ i costi dell'algoritmo INSERTION SORT nel caso peggiore e SELECTION SORT in quello medio, rispettivamente. Quale delle seguenti relazioni asintotiche è vera:
a) $f(n) = o(g(n))$ *b) $f(n) = \Theta(g(n))$ c) $f(n) = \omega(g(n))$ d) $g(n) = \omega(f(n))$
- Se $f(n) = n \log \sqrt{n}$ e $g(n) = n \log^2 n$, quale delle seguenti relazioni asintotiche è falsa:
a) $f(n) = o(g(n))$ b) $f(n) = O(g(n))$ c) $g(n) = \Omega(f(n))$ *d) $f(n) = \Theta(g(n))$
- Dato un problema con una delimitazione inferiore alla complessità temporale pari a $\Omega(f(n))$, un algoritmo per la sua risoluzione non può avere tempo di esecuzione $g(n)$ pari a:
a) $g(n) = \Theta(f(n))$ *b) $g(n) = o(f(n))$ c) $g(n) = \omega(f(n))$ d) $g(n) = O(f(n))$
- Nel caso medio, assumendo che le istanze siano equidistribuite, la ricerca di un elemento in un insieme non ordinato di n elementi richiede un numero di confronti pari a:
a) n b) $(n - 1)/2$ *c) $(n + 1)/2$ d) 1
- La delimitazione inferiore al problema dell'ordinamento ottenibile dagli alberi di decisione è:
a) $o(n \log n)$ b) $\omega(n \log n)$ *c) $\Theta(n \log n)$ d) $\Theta(n)$
- Siano $f(n)$ e $g(n)$ i costi degli algoritmi HEAPSORT e QUICKSORT, 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) $g(n) = \omega(f(n))$
- Sia dato un array A di n elementi in cui l'elemento massimo è pari a n^3 . Qual è la complessità temporale dell'algoritmo RADIX SORT applicato ad A ?
a) $\Theta(n^3)$ b) $\Theta(n \log_3 n)$ *c) $O(n)$ d) $\Theta(n \log n)$
- Una coda di priorità realizzata con un heap binario avente nella radice l'elemento minimo, supporta la ricerca del secondo elemento più piccolo in:
a) $\Theta(n \log n)$ b) $\Theta(n)$ c) $\Theta(\log n)$ *d) $O(1)$
- Un heap binomiale di 13 elementi è costituito dai seguenti alberi binomiali:
a) B_0, B_1, B_3 b) B_0, B_1, B_2 *c) B_0, B_2, B_3 d) B_1, B_2, B_3
- L'inserimento di un elemento in un dizionario di n elementi realizzato con un array ordinato costa:
a) $\Theta(\log n)$ b) $\Theta(1)$ c) $O(\log n)$ *d) $\Omega(n)$
- In un albero AVL di n elementi, la ricerca di un elemento nel caso migliore costa:
*a) $O(1)$ b) $\Theta(n)$ c) $\Theta(\log n)$ d) $\Theta(n/\log n)$
- Dati due elementi u, v appartenenti ad un universo totalmente ordinato U , una funzione hash $h(\cdot)$ si dice *perfetta* se:
a) $u = v \Rightarrow h(u) \neq h(v)$ b) $u \neq v \Rightarrow h(u) = h(v)$ c) $u = v \Rightarrow h(u) = h(v)$ *d) $u \neq v \Rightarrow h(u) \neq h(v)$
- Siano X e Y due stringhe di lunghezza m ed n . Qual è la complessità dell'algoritmo per la determinazione della distanza tra X e Y basato sulla tecnica della programmazione dinamica?
*a) $O(mn)$ b) $O(n)$ c) $O(m + n)$ d) $O(m)$

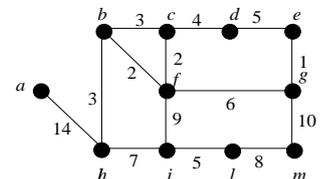


- Quanti archi contiene il sottografo indotto da $\{a, c, e\}$ nel seguente grafo?
a) 0 *b) 1 c) 2 d) 3

- La visita in ampiezza dell'albero di cui alla Domanda 15 eseguita partendo dal nodo c restituisce un BFS di altezza:
a) 1 *b) 2 c) 3 d) 4
- L'algoritmo di Bellman e Ford applicato ad un grafo pesato con un numero di archi $m = \Theta(n^2)$, ha complessità:
a) $\Theta(n^2)$ b) $\Theta(n + m)$ *c) $\Theta(n^3)$ d) $O(m \log n)$
- Dato un grafo pesato $G = (V, E)$ con n vertici ed $m > n$ archi, e presi 2 vertici u, v tali che l'arco $(u, v) \in E$ abbia peso minimo in G , trovare il cammino minimo tra u e v applicando l'algoritmo di Dijkstra che usa l'heap binario costa:
*a) $\Theta(n)$ b) $\Theta(m)$ c) $\Theta(1)$ d) $\Theta(m \log n)$
- Usando gli alberi *QuickUnion* e l'euristica dell'unione pesata *by size*, il problema della gestione di n insiemi disgiunti sottoposti ad $n - 1$ *Union* ed $m = n^2$ *Find* può essere risolto in:
a) $\Theta(n)$ b) $\Theta(n + m)$ c) $\Theta(n^2)$ *d) $O(n^2 \log n)$
- Dato un grafo connesso con n vertici ed m archi, l'algoritmo di Kruskal esegue un numero di operazioni *Union*(u, v) pari a:
a) $2m$ *b) $n - 1$ c) $\Theta(m \log n)$ d) $\Theta(\log n)$

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 Dijkstra sul seguente grafo, assumendo il nodo h quale nodo sorgente (ogni figura disegnata deve rappresentare lo stato dei nodi del grafo prima di visitare il prossimo nodo).