

Compito di Algoritmi e Strutture Dati

Corso di Laurea in Informatica

Appello 24 Novembre 2009

Domanda 1 – (15 punti)

Si dia la definizione di funzioni in relazione polinomiale tra loro e si provi o si confuti che le funzioni $f(n) = 2^n$ e $g(n) = 2^{2n}$ sono in relazione polinomiale tra loro.

Domanda 2 – (15 punti)

Si consideri un grafo diretto e pesato $G = (V, E)$, in cui ogni arco ha peso non-negativo. Si dia un algoritmo che calcola i cammini ottimi da un vertice sorgente a tutti gli altri vertici (Single Source Shortest Path Algorithm). Si analizzi l'algoritmo. (Il punteggio terrà conto dell'efficienza dell'algoritmo da voi proposto).

Domanda 3 – (5 punti)

Il vostro algoritmo non funzionerà per grafi diretti che hanno un ciclo negativo. Se ne fornisca una spiegazione.

Domanda 4 – (15 punti)

Si dia la definizione di macchina di Turing non-deterministica e si illustri come tale macchina può essere simulata da una macchina di Turing deterministica.

Domanda 5 – (15 punti)

Risolvere la seguente formula ricorsiva esattamente, ovvero, identificare la funzione $T(n)$ che la soddisfa. Verificare inoltre l'esattezza della soluzione trovata e mostrare che tale valore è in accordo con la soluzione fornita dal Master Theorem.

$$T(n) = \begin{cases} 2T(\frac{n}{2}) + n^3, & n > 2 \\ c, & \text{altrimenti} \end{cases}$$

Compito di Laboratorio di Algoritmi e Strutture Dati

Corso di Laurea in Informatica

Appello 25 Novembre 2009

Siano dati due insiemi numerici A e B ad elementi in \mathbb{N} rappresentati mediante alberi binari di ricerca con chiave numerica positiva (senza ripetizioni). Su di essi sono definite le seguenti operazioni:

1. $Membership(X, k)$: trova, se esiste, l'elemento k nell'insieme X ;
2. $Insert(X, k)$: inserisce un nuovo elemento k nell'insieme X , ottenendo un insieme ancora rappresentato da un albero binario di ricerca T' ;
3. $Unions(A, B)$: crea l'albero T' rappresentante l'unione dei due insiemi;
4. $Intersections(A, B)$: crea l'albero T' rappresentante l'intersezione dei due insiemi.

Si richiede un algoritmo in linguaggio C che consenta di effettuare ognuna delle operazioni sopra descritte in modo efficiente, utilizzando una struttura dati di tipo dinamico realizzata attraverso record e puntatori. Si richiede, inoltre, che gli elementi degli insiemi ottenuti come output delle procedure $Unions(A, B)$ e $Intersections(A, B)$ siano visualizzati in ordine crescente.

Commentare opportunamente il codice implementato, indicando i tempi di esecuzione delle procedure implementate.