008AA – ALGORITMICA E LABORATORIO
Verifica del 10 giugno 2009

Cognome Nome: **N. Matricola:** **Corso: A B**

Esercizio 1. *(2+3+6 punti)* Sia data una tabella hash T di m celle, le cui collisioni sono gestite con indirizzamento aperto e hash doppio $h(k, i)$, dove k è una chiave e i è l'indice del passo di probing.

(a) Descrivere che cosa è una *sequenza di probing* per una chiave k, e fornire una definizione per $h(k, i)$ nell'ipotesi di hash doppio con una dimensione della tabella pari a $m = 2^q$, dove q è un intero fissato.

(b) Fissato $m = 11$, si consideri l'hash doppio $h(k, i)$ con $h_1(k) = k \mod 11$ e $h_2(k) = 1+(k \mod 10)$. Si disegni il contenuto della tabella $T[0, 10]$ nell'ipotesi in cui si inseriscano le chiavi $\{5, 15, 2, 8, 30, 4, 19, 11, 26\}$, specificando per ogni chiave la sua sequenza di probing.

(c) Scrivere lo pseudocodice dell'algoritmo che inserisce una chiave k nella tabella $T[0, m - 1]$ assumendo di usare $h(k, i)$ per generare la sequenza di probing, e assumendo che le cancellazioni siano gestite in modo logico attraverso un vettore binario $B[0, m - 1]$ tale che $B[i] = 1$ se la cella $T[i]$ contiene una chiave cancellata *logicamente*.

Cognome Nome: **N.Matr:**

Esercizio 2. *(6+1 punti)* Dato un albero binario T, in cui ciascuno dei nodi u contiene un intero nel campo u.`dato`, diciamo che u è *selezionato* se u.`dato` è uguale alla somma degli interi contenuti nei suoi figli. Nel caso che un figlio di u sia $NULL$, si assuma che l'intero in esso *contenuto* sia 0. Progettare un algoritmo ricorsivo che conta il numero di nodi u in T che sono *selezionati*. Valutare anche la complessità in tempo al caso pessimo dell'algoritmo proposto.

Cognome Nome: **N.Matr:**

Esercizio 3. *(4+1+5+2 punti)* Sia dato un grafo orientato G, di n vertici e m archi. Progettare un algoritmo che:

(a) ricevuti in ingresso due vertici u, w e un intero positivo k, stabilisce se esiste un cammino che va da u a w attraversando al più k archi;

(b) valutare la complessità in tempo al caso pessimo dell'algoritmo proposto al punto (a);

(c) ricevuti in ingresso tre vertici u, v, w, stabilisce se esiste un cammino che va da u a w e **NON** passa da v;

(d) valutare la complessità in tempo al caso pessimo dell'algoritmo proposto al punto (c).