008AA – ALGORITMICA E LABORATORIO
Appello del 13 settembre 2010

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

Esercizio 1. *(3+3 punti)* Data la seguente funzione ricorsiva `foo`, trovare la corrispondente relazione di ricorrenza e risolverla utilizzando il teorema principale.

```
aux( n ){
   if ( n < 3 ) return 1;
   else {
      s = 0;
      while (n >=3){
         n = n/3;
         s = s + 1;
      }
      return s
   }
}

foo( n ){
   if ( n < 3 )
      return 1;
   else
      return aux( n ) * foo( n/3 ) * foo( n/3 )
}
```

Cognome Nome: **N.Matr:**

Esercizio 2. *(3+5 punti)* Sia data una tabella hash T di m celle, le cui collisioni sono gestite con indirizzamento aperto e scansione lineare $h(k, i) = (k + 7i)\ \%\ m$, dove $k > 0$ è una chiave e $i = 0, 1, 2, \ldots, m - 1$ è l'indice del passo di scansione.

(a) Disegnare il contenuto della tabella $T \equiv T[0, m - 1]$ (dove $m = 11$) nell'ipotesi in cui si inseriscano le chiavi $21, 33, 9, 24, 12, 27, 35$, specificando per ogni chiave la sua sequenza di posizioni così accedute (probing).

(b) Scrivere lo pseudocodice dell'algoritmo che inserisce una chiave $k > 0$ nella tabella T assumendo di usare $h(k, i)$ per generare la sequenza di posizioni accedute (probing), e assumendo che le chiavi con valore negativo indichino il fatto che sono state cancellate *logicamente* in precedenza.

Cognome Nome: **N.Matr:**

Esercizio 3. *(6+3 punti)* Un albero binario T si dice Δ-*bilanciato* se, per ogni nodo x, la differenza tra le altezze dei due figli di x (in valore assoluto) è al più Δ. Per $\Delta = 1$, otteniamo i classici alberi 1-bilanciati AVL. Si ipotizzi che $\Delta \geq 1$ e che T sia un albero binario i cui nodi contengono soltanto il puntatore al figlio sinistro e quello al figlio destro. Dati Δ e la radice di T in ingresso: (a) scrivere una funzione che stabilisce se T è Δ-bilanciato; (b) valutarne la complessità in tempo, motivando l'analisi dei costi.

Cognome Nome: **N.Matr:**

Esercizio 4. *(6+3 punti)* Una *componente connessa* di un grafo non orientato $G = (V, E)$ è un sottoinsieme massimale $S \subseteq V$ tale che, per ogni coppia di vertici in S, esiste sempre almeno un cammino in G che li collega. La dimensione della componente connessa S è data dal numero di vertici $|S|$ in essa contenuti. Per esempio, un grafo connesso ha una sola componente connessa $S = V$ di dimensione $|V|$. Un grafo in cui l'insieme E degli archi è vuoto, ha invece $|V|$ componenti connesse di dimensione pari a 1, dove ciascuna componente è formata da un singolo vertice. Dato G in ingresso, memorizzato mediante liste di adiacenza: (a) progettare un algortimo che calcola la dimensione della componente connessa più grande in G; (b) valutarne la complessità in tempo, motivando l'analisi dei costi.
Nota. Accompagnare l'eventuale pseudo-codice con una descrizione informale dell'idea algoritmica sottostante. Gli iscritti alla classe 26 possono descrivere l'algoritmo soltanto a parole.