Algoritmi e Strutture Dati
&
Laboratorio di Algoritmi e Programmazione

— Appello del 14 Gennaio 2009 —

Esercizio 1 - ASD

Si risolvano le seguenti ricorrenze, giustificando la risposta.

- $T(n) = 9T(\frac{n}{3}) + 3n^2 + 2\sqrt{n}$
- $T(n) = \frac{5}{2}T(\frac{n}{2}) + n$

Esercizio 2 - ASD

Date le seguenti procedure A e B, si determini la complessità asintotica della procedura A(n) su input n.

A(n)
1 $s \leftarrow 0$
2 **for** $i \leftarrow 1$ **to** n
3 **do** $s \leftarrow s +$ B(i)
4 **return** s

B(m)
1 $s \leftarrow 0$
2 **for** $j \leftarrow 1$ **to** m
3 **do** $s \leftarrow s + 1$
4 **return** s

Esercizio 3 - ASD

Scrivere un algoritmo che dato un albero binario di ricerca T, bilanciato e contenente chiavi tutte distinte, e due chiavi $k_1 < k_2$ restituisce **true** se e solo se T soddisfa anche la seguente proprietà:
- per ogni nodo x di T se $key[x] = k_1$ allora non esiste alcun nodo $y \neq x$ in T tale che $k_1 < key[y] < k_2$.
Dire qual è la complessità dell'algoritmo rispetto al numero delle chiavi memorizzate nell'albero e spiegare perché è corretto.

Esercizio 4 - ASD

Si disegni un albero R/B che contiene le seguenti chiavi: 6, 15, 8, 2, 24, 56, 3, 43, 12, 4, 7. Lo si trasformi poi in un albero 2-3-4.

Esercizio 5 - ASD

Si sviluppi un algoritmo che, dati un albero generale T e un intero positivo $k > 0$, conta il numero di nodi di grado k in T. Si supponga che l'albero generale sia rappresentato tramite gli attributi: `fratello` e `figlio`. (Ricordiamo che il grado di un nodo di un albero è pari al numero dei suoi figli.)

Esercizio 1 (Laboratorio)

Una sequenza ordinata è una collezione in cui gli elementi compaiono in modo ordinato e sono ammesse più copie dello stesso elemento. Si vuole realizzare una classe *SequenzaOrdinata* per rappresentare una sequenza ordinata di elementi di tipo stringa. La struttura dati scelta per memorizzare la sequenza ordinata è l'array.

```
public class SequenzaOrdinata {
    private static final int defaultSize = 100;
    private String[] S = new String[defaultSize];    // la sequenza e' un array
    private int count;     // tot. elementi della sequenza


    // pre:  s non nulla
    // post: aggiunge la stringa s alla sequenza ponendola nella
    //       posizione corretta rispetto all'ordine e aggiornando count.
    //       Ritorna true se l'operazione e' riuscita, false se non c'e' piu'
    //       spazio libero nell'array
    public boolean insert(String s) {...}

    ...
}
```

Si richiede di completare l'implementazione del metodo *insert* della classe *SequenzaOrdinata* riportata sopra e di dimostrarne la correttezza.

Esercizio 2 (Laboratorio)

1. Si consideri una tabella hash T[0,...,6] = [-, 1, C, -, 11, 19, 26] in cui le posizioni 0 e 3 sono libere e la posizione 2 risulta marcata in seguito ad un'operazione di cancellazione. La tabella è stata costruita utilizzando la funzione hash

   $$h(k) = k \bmod 7$$

   e tecnica di gestione delle collisioni ad indirizzamento aperto con scansione lineare. Si consideri il problema di inserire la chiave 67 nella tabella. Dire se la chiave viene inserita, eventualmente in quale posizione e quali posizioni della tabella vengono esaminate con insuccesso.

2. Si consideri una tabella hash T[0...22] in cui si vogliono memorizzare dati relativi agli studenti che sostengono questo appello d'esame. Gli studenti vengono identificati dal numero di matricola, che diventa la chiave della tabella hash. Si consideri la seguente funzione hash basata sul metodo del ripiegamento:

   $$h(K) = h(k_1k_2k_3k_4k_5k_6) = (k_1k_2 \ - \ k_3k_4 \ + \ k_5k_6) \bmod 23$$

   dove $k_1k_2k_3k_4k_5k_6$ sono le cifre decimali che compongono K. Ad esempio, h(816145) = (81 - 61 + 45) mod 23 = 65 mod 23 = 19. Partendo dalla tabella contenente solamente la chiave utilizzata come esempio (816145), si richiede di inserire in tabella le seguenti chiavi specificando per ciascuna la posizione di inserimento e il numero di accessi effettuati:

   (a) 837120
   (b) 829133
   (c) 818326
   (d) 792364

   Per la soluzione di eventuali collisioni utilizzare la scansione quadratica

   $$c(k,i) = (h(k) + c_1 \cdot i + c_2 \cdot i^2) \bmod 23 \quad 0 \leq i \leq 22$$

   con $c_1 = 1$ e $c_2 = 3$.