Algoritmi e Strutture Dati

Appello del 19 Marzo 2008

Modulo Teoria

Attenzione: Scrivere nome, cognome e numero di matricola sul foglio protocollo

Esercizio 1 [Punti 9]

Date le funzioni hash

$$h_1(k) = k \bmod 13 \text{ e } h_2(k) = (k \bmod 3) + 1,$$

si consideri una tabella hash di dimensione $m = 13$ e inizialmente vuota a indirizzamento aperto con hash doppio

$$h(k, i) = (h_1(k) + i \times h_2(k)) \bmod 13.$$

Mostrare il contenuto della tabella dopo l'inserimento delle chiavi $8, 3, 9, 4, 2, 16, 5$.

Esercizio 2 [Punti 12]

1. Progettare un algoritmo ricorsivo per trovare il minimo di un array A di n elementi.
2. Progettare un algoritmo ricorsivo per trovare simultaneamente il minimo e il massimo di un array A di n elementi.
3. Calcolare la complessitá temporale degli algoritmi proposti utilizzando le equazioni di ricorrenza.

Esercizio 3 [Punti 9]

Dato il grafo orientato $G = (V, E)$ di 8 vertici e 10 archi:

$$E = \{(a, b), (a, e), (b, c), (b, e), (c, a), (c, h), (d, e), (f, c), (f, g), (g, h)\},$$

1. dire se é possibile ordinare topologicamente G motivando la risposta. Nel caso di risposta negativa, si indichi una possibile modifica che renda G ordinabile topolgicamente. Calcolare l'ordine topologico di G o della sua modifica;

2. indicare la lista degli archi esaminati dalla visita BFS eseguita a partire dal vertice a e disegnare l'albero BFS;

3. indicare la lista degli archi esaminati dalla visita DFS eseguita a partire dal vertice a e disegnare l'albero DFS.

Modulo Laboratorio

Attenzione: Scrivere nome, cognome e numero di matricola sul foglio del testo e sul foglio protocollo

Esercizio 1 [Punti 15]

(Da svolgere su un foglio separato)

Si scriva in codice java l'implementazione ricorsiva di un metodo di visita post-ordine su un albero di ricerca binario, assumendo di avere la seguente classe BinaryTree (di cui omettiamo i dettagli implementativi):

```
public class BinaryTree {

    private BinaryNode radice; // Radice dell'albero


    public BinaryTree(){

        radice = new BinaryNode();
    }
    ...
}


public class BinaryNode {

    Object elemento;

    BinaryNode padre, figlioSx, figlioDx;


    public BinaryNode(){

        elemento = null;

        padre = figlioSx = figlioDx = null;
    }
    ...
}
```

Esercizio 2 [Punti 15]

(Inserire i commenti nel testo)

Si commenti il seguente codice con javadoc. Di quale algoritmo di ordinamento si tratta?

```
public static void Sort(Comparable a[]) {
    sort(a, 0, a.length-1);
}


private static void sort(Comparable a[], int lower, int upper){
    if (lower<upper){
        int pivotIndex = Partition(a, lower, upper);
        sort(a, lower, pivotIndex);
        sort(a, pivotIndex+1, upper);
    }
}


private static int Partition(Comparable a[], int lower, int upper){
    int half = (lower+upper)/2;
    Comparable pivot = a[half];
```

```
    a[half] = a[lower];

    a[lower] = pivot;

    Comparable temp = null;

    while (true){


        while (a[upper].compareTo(pivot)>0)

            upper--;


        while (a[lower].compareTo(pivot)<0)

            lower++;


        if (lower<upper){

            temp = a[lower];

            a[lower++] = a[upper];

            a[upper--] = temp;

        } else

            return upper;

    }
}
```