008AA – ALGORITMICA E LABORATORIO

Verifica del 6 giugno 2011

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

Esercizio 1. *(5+2 punti)* Scrivere un algoritmo ricorsivo `foo`(n) la cui complessità in tempo al caso pessimo sia modellata dalla seguente relazione di ricorrenza: $T(n) = O(1)$ per $n < 10$, e $T(n) = 3T(n/2) + \Theta(n^2)$ per $n \geq 10$. Infine si risolva anche la relazione suddetta.

Soluzione. È sufficiente progettare un algoritmo che consiste di due for innestati, ognuno esegue n passi, e al di fuori di questi due chiamate ricorsive a `foo(2n/3)`.

Per quanto riguarda la soluzione della relazione di ricorrenza, basta osservare che $a = 2, b = 3/2$ quindi $n^{\log_{3/2} 2} = O(n^{2-\epsilon}$ per ogni $0 < \epsilon \leq 2 - \log_{3/2} 2$. Ciò significa che siamo nel caso 1 del teorema master, e quindi la soluzione alla relazione di ricorrenza è $O(n^2)$.

Alternativamente, si può osservare che $\alpha = 2$, $\beta = 3/2$ e $f(n) = n^2$, per cui esiste una costante $\gamma = \frac{\alpha f(n/\beta)}{f(n)} = \frac{8}{9} < 1$: quindi vale il primo caso del teorema fondamentale sulle relazioni di ricorrenza, fornendo $T(n) = O(f(n)) = O(n^2)$.

Cognome Nome: **N.Matr:**

Esercizio 2. *(5 punti)* Si eseguano nell'ordine le seguenti operazioni su un max-heap H inizialmente vuoto:

```
Insert(1,H);    Insert(6,H);   Insert(8,H); Insert(4,H);    Insert(2,H);   Insert(9,H);
Delete_Max(H);
```

mostrando il contenuto di H (sia in forma di albero che di vettore) dopo l'esecuzione di ogni operazione.

Soluzione. Il contenuto del vettore prima della cancellazione è $H = [9, 4, 8, 1, 2, 6]$. Dopo la cancellazione del massimo risulta $H = [8, 4, 6, 1, 2]$.

Cognome Nome: **N.Matr:**

Esercizio 3. *(4 punti)* Sia dato un albero binario T di radice r, in cui ogni nodo u è etichettato con due interi $u.k$ e $u.n$. Progettare un algoritmo ricorsivo che, ricevuta in ingresso una coppia di valori (a, b), stampa il valore $u.k$ per quei nodi u tali che $a \leq u.k \leq b$ e il numero di discendenti di u in T (incluso u stesso) è pari a $u.n$. Si determini inoltre la complessità in tempo al caso pessimo dell'algoritmo proposto.

Soluzione. L'algoritmo deve prevedere il calcolo della dimensione di un sottoalbero da usare nel confronto con $u.n$.

```
int Check(u,a,b)
{
  if (u == NULL) return 0;
  L = Check(u.left, a, b);
  R = Check(u.right, a, b);
  if ( (a <= u.k <= b) && (u.n == L+R+1) ) print u.k;
  return (L+R+1);
}
```

L'algoritmo ha complessità $O(n)$, se $n = |T|$, e viene invocato come `Check(r,a,b)`.

Cognome Nome: **N.Matr:**

Esercizio 4. *(4+2 punti)* Sia data la sequenza di chiavi $S = \{7, 18, 19, 6, 5, 10\}$. Inserirle in un albero AVL inizialmente vuoto, indicando a ogni inserimento l'eventuale nodo critico, e l'operazione di ribilanciamento eseguita.

Infine indicare come avviene la cancellazione della chiave 7, e se questa sbilancia l'albero. **Soluzione.** Si tratta di una semplice simulazione.

Cognome Nome: **N.Matr:**

Esercizio 5. *(4+4 punti)* Sia dato il grafo orientato G formato da 6 vertici numerati da 1 a 6, e i seguenti archi orientati $E = \{(1,4;3),(2,1;4),(2,5;5),(3,1;5),(3,6;1),(4,2;5),(4,5;6),(4,6;1),(6,5;1)\}$, in cui la terza componente di ogni tripla denota il peso dell'arco specificato dalle prime due componenti. Si rappresenti il grafo orientato mediante liste di adiancenza ordinate in modo crescente secondo la numerazione dei vertici.

- Calcolare la visita DFS del grafo suddetto. Indicare l'albero DFS risultante e il tipo di ciascun arco non dell'albero.
- Calcolare l'albero dei cammini minimi con l'algoritmo di Dijkstra, a partire dalla sorgente 1. (Illustrare l'evoluzione della coda di priorità.) *[Non deve essere risolto dagli studenti della Classe 26.]*

Soluzione. Si tratta di semplici simulazioni.