Algoritmi e Strutture Dati 1

Appello del 26/05/2009

Prima parte

Compito n° 1

Esercizio 1

Dire se, giustificando la risposta,

a) $n \log n = \theta(n \log n + n)$
b) $n\sqrt{n} = \Omega(n \log n)$

Esercizio 2

Si supponga di dover ordinare un array di *n* numeri interi contenente numeri da 0 a $n \log \log n$. Si dica qual è la complessità, nel caso medio e nel caso peggiore, degli algoritmi di ordinamento

- Merge Sort
- Quick Sort
- Selection Sort
- Counting sort

Dire quindi, giustificando la risposta, qual è l'algoritmo più conveniente nel caso medio e quale nel caso peggiore.

Esercizio 3

Scrivere una procedura **P1** in pseudocodice che preso in input un array A di numeri interi, restituisce in output il numero massimo di volte per cui un elemento è ripetuto in A.

La procedura deve avere tempo di esecuzione nel caso peggiore $\theta(n \log n)$, e può sfruttare tutte le procedure viste a lezione (es: *InserctionSort, MergeSort, Quicksort, Binarysearch, LinearSearch, etc…*)

Esercizio 4

Sia $T(n) = \begin{cases} T(n/2) + \Theta(1) & se\ n > 1 \\ d & se\ n = 1 \end{cases}$

Si dia una stima esplicita (non ricorsiva) di T(n) facendo uso dei metodi di sostituzione e induzione.

Attenzione:

- Scrivere nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- non è ammesso per nessun motivo l'uso di telefoni cellulari, calcolatrici, etc…
- non è possibile consultare appunti, libri, dispense.

Algoritmi e Strutture Dati 1

Appello del 26/05/2009

Seconda parte

Compito n° 1

Esercizio 1

Si implementi in pseudocodice la strutture dati **O*rdered List,*** che consiste in una lista doppiamente collegata in cui gli elementi sono sempre ordinati in ordine crescente, e ogni intero non può essere presente più di una volta.

Si implementino per tale struttura dati le seguenti procedure:

- **Search** che prende in input una lista ordinata *OL* e un intero *x* e restituisce *true* se x è presente in OL, *false* altrimenti
- **Insert** che prende in input una lista ordinata *OL* e un intero *x*, e inserisce l'intero *x* in *OL* (se non è già presente)
- **Remove** che prende in input una lista ordinata *OL* e un intero *x*, e rimuove l'intero *x* da *OL* (se esso è presente)

Per ognuna delle procedure si stimi la complessità computazionale nel caso peggiore.

Esercizio 2

Si consideri una *tabella hash con lista di trabocco* contenente numeri interi.

- Si spieghi brevemente come una tale tabella possa essere implementata.

Supponendo che H(x) sia la funzione hash (già implementata) della tabella,

- scrivere una procedura **P3** che prende in input una tabella hash e restituisce il numero di elementi presenti nella più lunga lista di trabocco della tabella;
- scrivere una procedura **P4** che prende in input una tabella hash ed un intero *x* e, restituisce il più piccolo elemento maggiore di *x* presente in tabella (*NIL* se nessun elemento della tabella è maggiore di *x*).

Attenzione:

- Scrivere nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- non è ammesso per nessun motivo l'uso di telefoni cellulari, calcolatrici, etc…
- non è possibile consultare appunti, libri, dispense.

Algoritmi e Strutture Dati 1

Appello del 26/05/2009

Prima parte

Compito n° 2

Esercizio 1

Dire se, giustificando la risposta,

a) $n\sqrt{n} = \theta(n\sqrt{n} + n \ \ )$
b) $n \log n = \Omega(n)$

Esercizio 2

Si supponga di dover ordinare un array di *n* numeri interi contenente numeri da 0 a $n\sqrt{n}$. Si dica qual è la complessità, nel caso medio e nel caso peggiore, degli algoritmi di ordinamento

- Merge Sort
- Quick Sort
- Selection Sort
- Counting sort

Dire quindi, giustificando la risposta, qual è l'algoritmo più conveniente nel caso medio e quale nel caso peggiore.

Esercizio 3

Scrivere una procedura **P2** in pseudocodice che preso in input un array A di numeri interi, restituisce in output il numero minimo di volte per cui un elemento è ripetuto in A.

La procedura deve avere tempo di esecuzione nel caso peggiore $\theta(n \log n)$, e può sfruttare tutte le procedure viste a lezione (es: *InserctionSort, MergeSort, Quicksort, Binarysearch, LinearSearch, etc…*)

Esercizio 4

Sia $T(n) = \begin{cases} 2T(n/2) + \Theta(1) & se\ n > 1 \\ d & se\ n = 1 \end{cases}$

Si dia una stima esplicita (non ricorsiva) di T(n) facendo uso dei metodi di sostituzione e induzione.

Attenzione:

- Scrivere nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- non è ammesso per nessun motivo l'uso di telefoni cellulari, calcolatrici, etc…
- non è possibile consultare appunti, libri, dispense.

Algoritmi e Strutture Dati 1

Appello del 26/05/2009

Seconda parte

Compito n° 2

Esercizio 1

Si implementi in pseudocodice la strutture dati ***Reverse List,*** che consiste in una lista doppiamente collegata in cui gli elementi sono sempre ordinati in ordine decrescente, e ogni intero non può essere presente più di una volta.

Si implementino per tale struttura dati le seguenti procedure:

- **Find** che prende in input una Reverse List *RL* e un intero *x* e restituisce *true* se x è presente in RL, *false* altrimenti
- **Add** che prende in input una Reverse List *RL* e un intero *x*, e inserisce l'intero *x* in *RL* (se non è già presente)
- **Remove** che prende in input una Reverse List *RL* e un intero *x*, e rimuove l'intero *x* da *RL* (se esso è presente)

Per ognuna delle procedure si stimi la complessità computazionale nel caso peggiore.

Esercizio 2

Si consideri una *tabella hash con lista di trabocco* contenente numeri interi.

- Si spieghi brevemente come una tale tabella possa essere implementata.

Supponendo che H(x) sia la funzione hash (già implementata) della tabella,

- scrivere una procedura **P5** che prende in input una tabella hash e restituisce il numero di elementi presenti nella più corta lista di trabocco della tabella (non si tengano in considerazione la posizioni della tabella a cui sono associate liste di trabocco vuote);
- scrivere una procedura **P6** che prende in input una tabella hash ed un intero *x* e, restituisce il più grande elemento minore di *x* presente in tabella (*NIL* se nessun elemento della tabella è minore di *x*).

Attenzione:

- Scrivere nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- non è ammesso per nessun motivo l'uso di telefoni cellulari, calcolatrici, etc…
- non è possibile consultare appunti, libri, dispense.