

ESAME DI ALGORITMI E STRUTTURE DI DATI (A)
Lunedì 17 Giugno 2002

NOME:
COGNOME:
MATRICOLA:

Per un buon esito dell'esame, si consiglia di:

- scrivere in forma leggibile il proprio nome, cognome e matricola sul testo del compito e su ogni foglio consegnato;
- provare gli esercizi prima in brutta copia. Ricopiarli in bella copia e consegnare quest'ultima oltre al testo del compito;
- non fatevi prendere dal panico, e neppure dalla noia. Un giusto livello di tensione è ciò che serve;
- svolgere il compito individualmente. Passare copiando è dannoso per voi, e irrilevante per il docente.

Esercizio 1 (*Punti 18*)

1. *Si discutano brevemente le differenze tra le strutture di dati vettore e lista.*
2. *Una lista ordinata è una lista in cui gli oggetti sono ordinati in ordine crescente di chiave. Sia L una lista ordinata. Si scriva la procedura $SortedListInsert(L, x)$ che inserisce nella lista L , in posizione opportuna, l'oggetto puntato da x . Si calcoli la relativa complessità computazionale.*

Soluzione

Le differenze tra lista e vettore sono le seguenti: la lista è una struttura dinamica, cioè posso inserire e cancellare oggetti, modificandone la dimensione. Il vettore è invece una struttura statica. L'accesso agli oggetti in una lista è sequenziale; l'accesso agli elementi in un vettore è diretto. Infine, una lista è accessibile mediante puntatori che fanno riferimento agli oggetti della lista. Un vettore è accessibile mediante indici che fanno riferimento agli elementi del vettore.

Algoritmo 1 SortedListInsert(L,x)

SortedListInsert(L,x)

```
1:  $z \leftarrow \text{NIL}$ 
2:  $y \leftarrow \text{head}(L)$ 
3: while ( $y \neq \text{NIL}$ ) and ( $\text{key}[y] < \text{key}[x]$ ) do
4:    $z \leftarrow y$ 
5:    $y \leftarrow \text{next}[y]$ 
6: end while
7: if  $z \neq \text{NIL}$  then
8:    $\text{next}[z] \leftarrow x$ 
9:    $\text{prev}[x] \leftarrow z$ 
10: end if
11: if  $y \neq \text{NIL}$  then
12:    $\text{next}[x] \leftarrow y$ 
13:    $\text{prev}[y] \leftarrow x$ 
14: end if
15: if  $z = \text{NIL}$  then
16:    $\text{head}(L) \leftarrow x$ 
17: end if
```

La complessità è $\Theta(n)$, ove n è la lunghezza della lista.

Esercizio 2 (Punti 12)

La Signora Edda è giunta al seguente paradosso. Ella sostiene di aver appena scoperto il seguente algoritmo di ordinamento per confronto con complessità pessima lineare: dato un albero binario di ricerca, si visiti tale albero in ordine intermedio (sinistra-radice-destra), salvando le chiavi dei nodi visitati. Al termine della visita, le chiavi saranno state salvate in ordine crescente. Tale algoritmo ha complessità pessima lineare $\Theta(n)$, ove n è il numero di nodi dell'albero. D'altronde, ella è a conoscenza dell'esistenza di un teorema che afferma che il problema dell'ordinamento per confronto ha complessità pessima pari a $\Theta(n \log n)$, ove n è il numero di elementi da ordinare. Dunque, non possono esistere algoritmi di ordinamento per confronto con complessità pessima lineare. Dove sbaglia la Signora Edda?

Soluzione

La Signora Edda sbaglia in quanto l'algoritmo che lei sostiene di aver trovato risolve solo parzialmente il problema dell'ordinamento. Infatti, il problema dell'ordinamento consiste nel trovare una permutazione ordinata di una sequenza fornita in ingresso. L'input del problema dell'ordinamento è dunque una sequenza, e non un albero binario di ricerca, in cui le chiavi hanno una certa organizzazione (secondo la proprietà degli alberi binari di ricerca). Dunque l'algoritmo è parziale in quanto manca un passo iniziale che costruisce un albero binario di ricerca a partire da una sequenza. Questo passo ha complessità pessima pari a $\Theta(n \log n)$, ove n è la lunghezza della sequenza.

ESAME DI ALGORITMI E STRUTTURE DI DATI (B)
Lunedì 17 Giugno 2002

NOME:
COGNOME:
MATRICOLA:

Per un buon esito dell'esame, si consiglia di:

- scrivere in forma leggibile il proprio nome, cognome e matricola sul testo del compito e su ogni foglio consegnato;
- provare gli esercizi prima in brutta copia. Ricopiarli in bella copia e consegnare solo quest'ultima;
- non fatevi prendere dal panico, e neppure dalla noia. Un giusto livello di tensione è ciò che serve;
- svolgere il compito individualmente. Passare copiando è dannoso per voi, e irrilevante per il docente.

Esercizio 3 (*Punti 18*)

1. *Si discutano brevemente le differenze tra le strutture di dati vettore e lista.*
2. *Una lista ordinata è una lista in cui gli oggetti sono ordinati in ordine crescente di chiave. Sia L una lista ordinata. Si scriva la procedura $\text{SortedListSearch}(L, k)$ che cerca nella lista L l'oggetto con chiave k e ritorna un puntatore a tale oggetto, se esso esiste, e ritorna NIL altrimenti. Si calcoli la relativa complessità computazionale.*

Soluzione

Le differenze tra lista e vettore sono le seguenti: la lista è una struttura dinamica, cioè posso inserire e cancellare oggetti, modificandone la dimensione. Il vettore è invece una struttura statica. L'accesso agli oggetti in una lista è sequenziale; l'accesso agli elementi in un vettore è diretto. Infine, una lista è accessibile mediante puntatori che fanno riferimento agli oggetti della lista. Un vettore è accessibile mediante indici che fanno riferimento agli elementi del vettore.

Algoritmo 2 SortedListSearch(L,k)

SortedListSearch(L,k)

```
1:  $x \leftarrow \text{head}(L)$ 
2: while ( $x \neq \text{NIL}$ ) and ( $\text{key}[x] < k$ ) do
3:    $x \leftarrow \text{next}[x]$ 
4: end while
5: if ( $x = \text{NIL}$ ) or ( $\text{key}[x] > k$ ) then
6:   return NIL
7: else
8:   return  $x$ 
9: end if
```

La complessità è $\Theta(n)$, con n la lunghezza della lista.

Esercizio 4 (Punti 12)

La Signora Edda è giunta al seguente paradosso. Ella sostiene di aver appena scoperto il seguente algoritmo di ordinamento per confronto con complessità pessima lineare: dato un albero binario di ricerca, si visiti tale albero in ordine intermedio (sinistra-radice-destra), salvando le chiavi dei nodi visitati. Al termine della visita, le chiavi saranno state salvate in ordine crescente. Tale algoritmo ha complessità pessima lineare $\Theta(n)$, ove n è il numero di nodi dell'albero. D'altronde, ella è a conoscenza dell'esistenza di un teorema che afferma che il problema dell'ordinamento per confronto ha complessità pessima pari a $\Theta(n \log n)$, ove n è il numero di elementi da ordinare. Dunque, non possono esistere algoritmi di ordinamento per confronto con complessità pessima lineare. Dove sbaglia la Signora Edda?

Soluzione

La Signora Edda sbaglia in quanto l'algoritmo che lei sostiene di aver trovato risolve solo parzialmente il problema dell'ordinamento. Infatti, il problema dell'ordinamento consiste nel trovare una permutazione ordinata di una sequenza fornita in ingresso. L'input del problema dell'ordinamento è dunque una sequenza, e non un albero binario di ricerca, in cui le chiavi hanno una certa organizzazione (secondo la proprietà degli alberi binari di ricerca). Dunque l'algoritmo è parziale in quanto manca un passo iniziale che costruisce un albero binario di ricerca a partire da una sequenza. Questo passo ha complessità pessima pari a $\Theta(n \log n)$, ove n è la lunghezza della sequenza.