Algoritmi e Strutture Dati - Scienze di Internet
Esercitazione 3

Gabriele D'Angelo
<gdangelo@cs.unibo.it>

19/10/2004

1 Esercizio sulle liste I

Si consideri una nuova operazione delle liste che, data una lista L di interi, la modifica duplicando tutti gli elementi divisibili per 3 mantenendo lo stesso ordine che gli elementi avevano in L (p.e. se l'ingresso è L=6,1,3,4, allora il risultato è L=6,6,1,3,3,4). Si scriva una procedura Pascal di complessità ottima utilizzando gli operatori per le liste visti a lezione.

Soluzione:

```
procedure DUPLICA3(var L : lista);
var p : posizione;
    a : integer;
Begin
   p := PRIMOLISTA(L);
   while not FINELISTA(p, L) do Begin
      a := LEGGILISTA(p, L);
      if (a mod 3) = 0 then begin
         p := SUCCLISTA(p, L);
         INSLISTA(a, p, L)
      end;
      p := SUCCLISTA(p, L)
    end
End;
```

per il codice completo fare riferimento al file duplica3.pas in codice3.tgz

2 Esercizio sulle liste II

Si consideri una nuova operazione delle liste che, data una lista L di interi, la modifica cancellando tutti gli elementi pari da L e copiando tutti quelli dispari in un'altra lista M, mantenendo sia in L che in M l'ordine che gli elementi avevano originariamente in L (p.e. se l'ingresso è L=4,1,6,3, allora il risultato è L=1,3 ed M=1,3). Si scriva una procedura Pascal di complessità ottima utilizzando gli operatori per le liste visti a lezione.

Soluzione:

```
procedure PARIDISPARI(var L, M : lista);
var p, q : posizione;
Begin
   p := PRIMOLISTA(L);
   CREALISTA(M);
   q := PRIMOLISTA(M);
   while not FINELISTA(p, L) do
      if (LEGGILISTA(p, L) mod 2 = 0) then
         CANCLISTA(p, L)
      else Begin
         INSLISTA(LEGGILISTA(p, L), q, M);
         p := SUCCLISTA(p, L);
         q := SUCCLISTA(q, M)
      end
End;
```

per il codice completo fare riferimento al file paridispari.pas in codice3.tgz

3 Esercizio sulle liste III

L'esercizio 2.6 a pagina 47 del libro di testo richiede:

"Si scriva una procedura Pascal ricorsiva di complessità ottima che, data una lista L di interi, riordina L in modo che tutti gli elementi dispari precedano nello stesso ordine che avevano inizialmente in L, tutti gli elementi pari (p.e., se L=3, 7, 8, 1, 4, allora si ottiene L=3, 7, 1, 8, 4).

Soluzione

dal testo si nota chiaramente come vi siano vari requisiti:

1. complessità ottima
2. riordino della lista L
3. procedura ricorsiva

E' abbastanza evidente come la complessità ottima in questo caso sia O(n), ovvero fatto salvo costanti moltiplicative il costo computazionale è lineare rispetto alla dimensione dell'input, ancora più evidente è come ogni elemento debba essere vagliato almeno una volta visto che è necessario verificare se è pari o dispari.

Il requisito 2 lascia supporre che la soluzione banale (scorrere la lista e creare due altre liste, una contenente i numeri dispari ed una i pari, per poi fonderle in un'unica lista) non sia considerata pienamente valida.

Tralasciando per un momento il requisito 3 vediamo (facendo riferimento al sorgente *paredis.pas* contenuto in codice3.tgz) come sia possibile risolvere l'esercizio procedendo in questo modo:

1. la lista viene scorsa dalla testa fino alla coda
2. per ogni elemento si verifica se questo è pari o dispari

3. nel caso sia dispari non è necessario fare nulla

4. nel caso invece sia pari si procede alla sua eliminazione dalla posizione attuale e al ricollocamento in coda alla lista

5. il ciclo precedente deve continuare fino a quando non sia stata scorsa tutta la lista, così da preservare l'ordinamento interno delle sequenze pari e dispari

Come preannunciato il costo computazionale della soluzione è O(n), gli elementi in input vengono vagliati tutti una sola volta: si ricorda che in questa notazione non vengono prese in considerazione eventuali costanti moltiplicative. Attraverso un codice più complesso sarebbe possibile realizzare varie ottimizzazioni, utili in casi particolari ma che non cambierebbero l'ordine computazionale della soluzione.

Riprendendo la condizione 3 nel codice *paredisric.pas* contenuto in codice3.tgz troviamo una soluzione ricorsiva all'esercizio, non si è fatto altro che rendere in forma ricorsiva il ciclo di analisi della lista, si nota come questa soluzione appaia forzata e non particolarmente elegante.