
Esercitazione 2

Algorithmi e Strutture Dati (Informatica)

A.A 2015/2016

Tong Liu

March 14, 2016

Soluzioni esercizi

Esercizio 1

[Libro 3.7 Variante] Si scriva una procedura che, data una coda L di interi genera un'altra coda L' in modo che tutti gli elementi dispari precedano, nello stesso ordine che avevano inizialmente in L , tutti gli elementi pari (per esempio, se $L = 3, 7, 8, 1, 4$, allora si ottiene $L' = 3, 7, 1, 8, 4$),

Soluzione

Soluzione mostrata nell'Algoritmo 1

Esercizio 2

Un compito comune del compilatore e l'editore codice è determinare se le parentesi sono bilanciate e nestate bene, per esempio la stringa, $((()())()$ le parentesi sono bilanciate, invece $)()()$ e $()()$ non sono. Proporre un algoritmo che adopera la struttura pila e che ritorna true se le parentesi sono bilanciate e false altrimenti, se possibile, ritorna anche la posizione della parentesi che ha violato il bilanciamento. Input un array di char.

Soluzione

Soluzione mostrata nell'Algoritmo 2. Un vostro collega a lezione ha proposto una soluzione più efficiente, cioè $\text{if } c = '(' \text{, } P.\text{push}(1) \text{, else if } c = ')' \text{, } P.\text{pop}()$. In questo modo, evita di memorizzare tutte le char inutili.

Algorithm 1 Dispari-precedono

```
1: procedure DISPARI-PRECEDONO(QUEUE  $L$ )
2:   QUEUE  $L_1 \leftarrow$  QUEUE()
3:   QUEUE  $L_2 \leftarrow$  QUEUE()
4:   QUEUE  $L' \leftarrow$  QUEUE()
5:   while not  $L$ .isEmpty() do
6:     Integer  $a \leftarrow L$ .dequeue()
7:     if  $a$ .isDispari then
8:        $L_1$ .enqueue( $a$ )
9:     else
10:       $L_2$ .enqueue( $a$ )
11:    end if
12:  end while
13:  while not  $L_1$ .isEmpty() do
14:     $L'$ .enqueue( $L_1$ .dequeue())
15:  end while
16:  while not  $L_2$ .isEmpty() do
17:     $L'$ .enqueue( $L_2$ .dequeue())
18:  end while
19: end procedure
```

Algorithm 2 Verifica Bilanciamento

```
1: procedure VERIFICA-BILANCIAMENTO(Char[]  $S$ , Integer  $n$ )
2:   Pila  $P \leftarrow$  Pila()
3:   for Integer  $i \leftarrow 0$  to  $n - 1$  do
4:     char  $c \leftarrow S[i]$ 
5:     if  $c$  is not ')' then
6:        $P$ .push( $c$ )
7:     else if  $c$  is '(' then
8:       Char  $t \leftarrow "$ 
9:       while  $t$  is not '(' do
10:         $t \leftarrow P$ .pop()
11:       if  $P$ .isEmpty() then
12:         return false
13:       end if
14:     end while
15:   end if
16: end for
17: return true
18: end procedure
```

Esercizio 3

[2/2014] Data una lista L di interi si vuole togliere da L ogni elemento pari e inserirlo in una nuova lista M , mantenendo in entrambe le liste l'ordine originario degli elementi.

Soluzione

Soluzione mostrata nell'Algoritmo 3

Algorithm 3 Pari-dispari

```
1: procedure PARI-DISPARI(List  $L$ )
2:   List  $M \leftarrow$  List()
3:   Pos  $p \leftarrow L.head()$ 
4:   Pos  $q \leftarrow M.head()$ 
5:   while not  $L.finished(p)$  do
6:     Integer  $v \leftarrow L.read(p)$ 
7:     if  $v.is\ pari$  then
8:        $M.insert(q, v)$ 
9:        $L.remove(p)$ 
10:       $q \leftarrow M.next(q)$ 
11:     else
12:        $p \leftarrow L.next(p)$ 
13:     end if
14:   end while
15:   return  $M$ 
16: end procedure
```

Esercizio 4

[1/2014] Data una lista L interi si vuole modificarla cancellando tutti gli elementi adiacenti che hanno valori consecutivi mantenendo l'ordine iniziale degli elementi (p.e. se l'ingresso è 4,5,6,2,7,3,2,1,2,5 allora il risultato è 2,7,5).

Soluzione

Soluzione mostrata nell'Algoritmo 4

Esercizio 5

Il rango di un elemento di una lista di interi è la somma degli elementi successivi più se stesso. Rango di [3,2,5] è [10,7,5]. Creare una funzione che calcola il rango di una lista.

Algorithm 4 Cancella-Adiacenti

```
1: procedure CANCELLA-ADIANCENTI(List  $L$ )
2:   Pos  $p \leftarrow L.head()$ 
3:   Integer  $c \leftarrow \infty$ 
4:   while not  $L.finished(p)$  do
5:     Pos  $p' \leftarrow L.next(p)$ 
6:     Integer  $v_1 \leftarrow L.read(p)$ 
7:     Integer  $v_2 \leftarrow L.read(p')$ 
8:     if  $|v_1 - v_2| = 1$  then
9:        $c \leftarrow v_1$ 
10:       $L.remove(p)$ 
11:    else if  $|v_1 - c| = 1$  then
12:       $L.remove(p)$ 
13:    end if
14:     $p \leftarrow p'$ 
15:  end while
16: end procedure
```

Soluzione

Versione Iterativa mostrata in Pseudocodice 5 (Solo per l'allenamento del ragionamento, poco consigliato in pratica). Versione ricorsiva (consigliato) stà nell'esercizio 3.8 della Pagina 68 del libro.

Algorithm 5 Rango-Iterativo

```
1: procedure RANGO-ITERATIVO(List  $L$ , Pos  $p$ )
2:   Pos  $p \leftarrow L.head()$ 
3:   Integer  $i \leftarrow 1$ 
4:   while not  $L.finished(p)$  do
5:     Pos  $p \leftarrow L.next(p)$ 
6:      $i \leftarrow i + 1$ 
7:   end while
8:   for  $i$  downto 0 do
9:     Integer  $j \leftarrow i$ 
10:    Pos  $r \leftarrow L.head()$ 
11:    for  $j$  downto 2 do
12:       $r \leftarrow L.next(r)$ 
13:    end for
14:    Integer  $v \leftarrow L.read(r) + L.read(L.next(r))$ 
15:     $L.write(r, v)$ 
16:  end for
17: end procedure
```
