

Corso di Programmazione

Esame del 17 Settembre 2004

cognome e nome

Risolvi i seguenti esercizi e riporta le soluzioni in modo chiaro, giustificandole sinteticamente.

1. Procedure in Scheme

Definisci una procedura *process* in Scheme che, data una funzione $f : D \times D \rightarrow D$ che rappresenta un'operazione binaria e data una lista non vuota $(x_1 \ x_2 \ x_3 \ \dots \ x_{n-1} \ x_n)$ di elementi di D , permetta di calcolare il valore della seguente espressione: $f(f(\dots f(f(x_1, x_2), x_3), \dots x_{n-1}), x_n)$. Nel caso in cui la lista contenga un solo elemento, il valore restituito è costituito da quell'elemento.

2. Dimostrazioni per induzione

Con riferimento all'esercizio precedente, dimostra per induzione che la seguente proprietà vale per tutti gli n naturali:

$$(process \ (+ \ '(1 \ 2 \ \dots \ n - 1 \ n))) \longrightarrow \frac{n \cdot (n + 1)}{2}$$

In particolare:

- Scrivi formalmente la proprietà che intendi dimostrare per induzione.
- Scrivi formalmente la proprietà che esprime il caso base.
- Scrivi formalmente l'ipotesi induttiva.
- Scrivi formalmente la proprietà che si deve dimostrare come passo induttivo.
- Dimostra formalmente il caso base.
- Dimostra formalmente il passo induttivo.

3. Scheme e Java

Traduci la seguente procedura Scheme in un corrispondente metodo statico nel linguaggio Java, basato sulla stessa struttura ricorsiva.

```
(define manh3
  (lambda (i j k) ; i, j, k >= 0 interi
    (cond ((and (> i 0) (> j 0) (> k 0))
      (+ (manh3 (- i 1) j k) (manh3 i (- j 1) k) (manh3 i j (- k 1)))))
      ((and (> i 0) (> j 0))
      (+ (manh3 (- i 1) j 0) (manh3 i (- j 1) 0)))
      ((and (> i 0) (> k 0))
      (+ (manh3 (- i 1) 0 k) (manh3 i 0 (- k 1)))))
      ((and (> j 0) (> k 0))
      (+ (manh3 0 (- j 1) k) (manh3 0 j (- k 1)))))
      (else 1)
    )
  )
)
```

4. Memoization

Realizza un programma in Java che applichi la tecnica di memoization per trasformare la soluzione ricorsiva dell'esercizio precedente.

5. Classi in Java

La classe *ExtendedQueue* permette di istanziare code di numeri reali (*float*) per le quali è definito il seguente protocollo:

- il costruttore crea una coda vuota;
- il metodo *length()* permette di conoscere la lunghezza (il numero di elementi) della coda;
- *enqueue(x)* aggiunge il numero reale x in fondo alla coda;
- *dequeue(k)* toglie dalla coda i primi k elementi;
- infine, *get(i)* restituisce l'elemento che occupa l' i -ma posizione nella coda (*get(0)* è il primo elemento).

Proponi una realizzazione della classe *ExtendedQueue*, formalizzando in Java lo stato interno e il corpo del costruttore e di ciascuno dei metodi.