

Corso di Programmazione

Esame del 3 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 $(x_n x_{n-1} \dots x_3 x_2 x_1)$ di elementi di D , permetta di calcolare il valore della seguente espressione: $f(x_n, f(x_{n-1}, \dots f(x_3, f(x_2, x_1)) \dots))$. 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:

$$(process \ (n \ n - 1 \ \dots \ 2 \ 1)) \longrightarrow \left\lceil \frac{n}{2} \right\rceil$$

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 N
  (lambda (h b) ; h, b >= 0 interi
    (if (<= h b)
        (+ h 1)
        (+ 1 (+ (N (- h 1) b) (N (- h (+ b 1)) b)))
    )
  )
)
```

4. Programmazione dinamica

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

5. Classi in Java

La classe *ExtendedStack* permette di istanziare pile di numeri interi per le quali è definito il seguente protocollo:

- il costruttore crea una pila vuota;
- il metodo *height()* permette di conoscere l'altezza (il numero di elementi) della pila;
- *push(x)* aggiunge l'intero x in cima alla pila;
- *pop(k)* toglie dalla pila i k elementi più in alto;
- infine, *get(i)* restituisce l'elemento che si trova i posizioni sotto la cima (*get(0)* è l'elemento in cima).

Per esempio, per eseguire una divisione secondo il tradizionale modello in cui gli operandi si trovano in cima alla pila S e devono essere sostituiti dal risultato, si può utilizzare il seguente codice Java:

```
if (S.height() >= 2) { int y = S.get(1) / S.get(0); S.pop(2); S.push(y); }
```

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