



## ***Esercitazione di laboratorio n. 9***

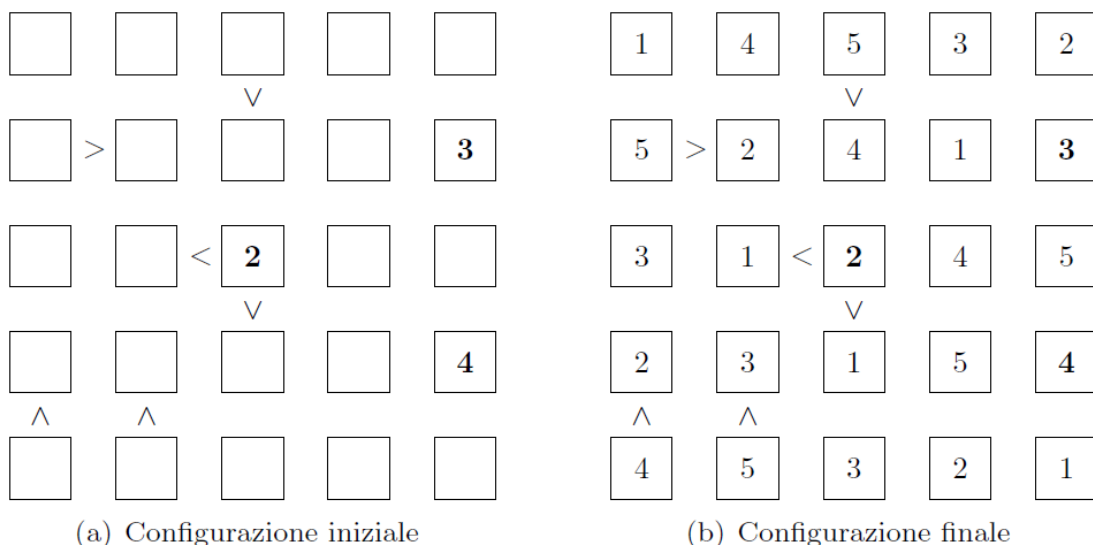
**Laboratorio valutato nei giorni 16/12/2015 (SQ1 e SQ2) e 18/12/2015 (SQ3 e SQ4)**  
**Caricamento nella Sezione Elaborati del Portale entro e non oltre le 23:59 del 22/12/2015**

### **Esercizio n. 1:** Futoshiki

Il Futoshiki (disuguaglianza in giapponese) è un gioco logico-matematico il cui obiettivo è riempire una griglia di dimensione  $n \times n$  con numeri nell'intervallo  $[1 \dots n]$ , rispettando le seguenti condizioni:

- in nessuna riga e in nessuna colonna una data cifra può essere ripetuta
- le cifre devono rispettare eventuali vincoli di disuguaglianza, se presenti
- eventuali cifre già inserite nella configurazione iniziale della griglia non possono essere modificate.

La figura seguente mostra una possibile configurazione iniziale e una soluzione valida per la stessa.



Lo scopo del gioco è riempire le caselle vuote con cifre nell'intervallo ammesso rispettando tutte le regole di cui sopra.

Si scriva un programma C che, dopo aver letto da file la configurazione iniziale, generi automaticamente una soluzione o tutte le soluzioni a scelta dell'utente.

La configurazione iniziale è riportata in un file (`futoshiki.txt`), il cui formato è il seguente:

- sulla prima riga è riportato il valore  $n$
- seguono  $n \times n$  righe, associate ognuna a una cella della tavola di gioco (da sinistra verso destra e dall'alto verso il basso) riportanti ognuna 5 interi  $V \ N \ O \ S \ E$ .
  - $V$  è il numero associato alla cella corrente. Se la cella è vuota,  $V$  vale 0
  - $N \ O \ S \ E$  rappresentano i quattro punti cardinali (in senso antiorario dall'alto) rispetto alla cella corrente, per indicare eventuali segni di



disuguaglianza. Al simbolo  $>$  è associato 1, al simbolo  $<$  è associato -1 e a simbolo assente è associato 0.

In relazione all'esempio, le prime 10 righe del file `futoshiki.txt` contengono:

```
5
0 0 0 0 0
0 0 0 0 0
0 0 0 1 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 1
0 0 -1 0 0
0 -1 0 0 0
0 0 0 0 0
3 0 0 0 0
```

### **Esercizio n. 2:** Catalogo di prodotti (BST)

Si realizzi un programma C che, attraverso un'apposita interfaccia utente, permetta di gestire una struttura dati di tipo BST (albero binario di ricerca) per la memorizzazione e l'interrogazione di un catalogo di prodotti. Ogni prodotto sia caratterizzato da un prezzo (float) e da un nome (stringa di lunghezza massima pari a 20 caratteri).

Le operazioni permesse devono essere quelle di:

- creazione di un nuovo BST (vuoto)
- inserimento in foglia di un nuovo elemento nel BST
- inserimento in radice di un nuovo elemento nel BST
- ricerca di un elemento (specificato dall'utente, per nome) nel BST
- stampa dell'elemento a prezzo minimo (massimo) presenti nel BST
- visualizzazione (a video) di tutti gli elementi presenti nel BST, alternativamente secondo pre/post/in order (a scelta dell'utente)
- calcolo del numero totale di nodi nell'albero
- salvataggio del BST su file (elencando i nodi in un ordine qualsiasi)
- caricamento di un nuovo BST da mediante una sequenza di inserzioni in foglia. Ogni riga del file contiene le informazioni relative al nodo da inserire nella forma `prezzo nome_prod.`

Il programma deve essere realizzato su tre moduli distinti:

- l'interfaccia utente
- un modulo per l'ADT di prima categoria BST
- un modulo per l'ADT di prima categoria del tipo Item per i prodotti, da usarsi come informazione memorizzata nei nodi dell'albero.

Sia il prezzo dell'Item prodotto la chiave su cui basare l'ordinamento del BST.