



## ***Esercitazione di laboratorio n. 8***

**La corretta risoluzione degli esercizi 1 e 2 sarà oggetto di valutazione ai fini dell'attribuzione del punteggio per i laboratori.**

### **Esercizio n. 1:** fuoriordine

Sia dato un vettore dinamico A di N elementi distinti contenente dati interi nell'intervallo  $[0 .. N-1]$ . La dimensione N sia passata al programma sulla riga di comando. I dati siano acquisiti da tastiera.

Si scriva un programma ricorsivo che conteggi quante sono le coppie (i, j), dove  $i, j \in [0 .. N-1]$  tali che

$$i < j \ \&\& \ A[i] > A[j]$$

### **Esempio:**

Si ipotizzi  $N=8$  e che i dati siano

4 6 0 1 2 5 3 7

Il programma deve concludere che le coppie sono 10. Pur non richiesto, a titolo chiarificatore si riporta l'elenco di tali coppie:

(4, 0), (4, 1), (4, 2), (4, 3), (6, 0), (6, 1), (6, 2), (6, 3), (6, 5), (5, 3).

### **Vincoli:**

**non verranno considerate soluzioni non ricorsive di complessità quadratica.**

### **Suggerimento:**

Ci si ispiri al mergesort per il processo di divisione ricorsiva e ad una opportuna variante di merge per la ricombinazione delle soluzioni. Ad ogni chiamata ricorsiva a merge, la selezione di un dato dal sottovettore destro comporta l'individuazione di un numero di coppie disordinate legato alla parte residua del sottovettore sinistro.

### **Esercizio n. 2:** forniture di medicinali

Un ospedale deve rifornire la sua farmacia interna con prodotti che sono caratterizzati da un terna di informazioni (ciascuna rappresentata da una stringa di non più di 20 caratteri):

categoria farmaceutica      nome del prodotto      ditta produttrice

L'elenco delle informazioni di cui sopra viene letto da un file di testo la cui lunghezza non è nota a priori. Si può comunque assumere che non vi siano più di 20 categorie farmaceutiche, 50 prodotti e 10 ditte produttrici. Il file non è ordinato.

Si scriva un programma ricorsivo che identifichi **tutte** le possibili forniture, cioè n-uple di coppie (prodotto, ditta) che soddisfano i seguenti vincoli:

- n è il numero di categorie farmaceutiche distinte, quindi nella n-upla sono considerate tutte le categorie farmaceutiche
- per ogni categoria nella n-upla si seleziona una sola coppia (prodotto, ditta)
- in ogni n-upla ogni ditta non fornisce all'ospedale più di k prodotti, dove k è un intero passato al programma sulla riga di comando.

Il programma può eventualmente concludere che la soluzione che rispetta i vincoli di cui sopra sui dati forniti non esiste.



**Esempio:**

Il file contiene le seguenti righe:

Antiipertensivi	Doxazosina	Teva
Antiipertensivi	Doxazosina	Mylan
Antiacidi	Alka-Seltzer	Bayer
Antiipertensivi	Cardura	Pfizer
Antiacidi	Magnesia	Pfizer
Antipiretici	Zerinol	Boehringer
Antipiretici	Paracetamolo	Mylan
Antiipertensivi	Catapresan	Boehringer
Antipiretici	Aspirina	Bayer

Si ipotizzi  $k=1$ , cioè i prodotti presenti in ciascuna fornitura provengono tutti da ditte diverse. Nell'esempio il numero di categorie farmaceutiche distinte è  $n=3$ . Sono possibili le seguenti 13 soluzioni, cioè terne (antipiretico, antiipertensivo, antiacido), dove per ognuna delle 3 categorie è stato individuato 1 solo prodotto (e la relativa ditta produttrice) e in ciascuna terna ogni ditta non compare più di  $k=1$  volte:

<b>Antiipertensivi</b>	<b>Antiacidi</b>	<b>Antipiretici</b>
Doxazosina (Teva)	Alka-Seltzer (Bayer)	Zerinol (Boehringer)
Doxazosina (Teva)	Alka-Seltzer (Bayer)	Paracetamolo (Mylan)
Doxazosina (Teva)	Magnesia (Pfizer)	Zerinol (Boehringer)
Doxazosina (Teva)	Magnesia (Pfizer)	Paracetamolo (Mylan)
Doxazosina (Teva)	Magnesia (Pfizer)	Aspirina (Bayer)
Doxazosina (Mylan)	Alka-Seltzer (Bayer)	Zerinol (Boehringer)
Doxazosina (Mylan)	Magnesia (Pfizer)	Zerinol (Boehringer)
Doxazosina (Mylan)	Magnesia (Pfizer)	Aspirina (Bayer)
Cardura (Pfizer)	Alka-Seltzer (Bayer)	Zerinol (Boehringer)
Cardura (Pfizer)	Alka-Seltzer (Bayer)	Paracetamolo (Mylan)
Catapresan (Boehringer)	Alka-Seltzer (Bayer)	Paracetamolo (Mylan)
Catapresan (Boehringer)	Magnesia (Pfizer)	Paracetamolo (Mylan)
Catapresan (Boehringer)	Magnesia (Pfizer)	Aspirina (Bayer)

L'ordine di elencazione delle n-uple risultato è irrilevante.

**Suggerimento:**

- si proceda per enumerazione delle possibilità. Si noti che è possibile, anche se non richiesto, effettuare operazioni di pruning



- si disaccoppino i nomi (delle categorie, dei prodotti e delle ditte) dalla funzione ricorsiva, definendo un opportuno ADT di I categoria Tabella di Simboli che permetta una corrispondenza nomi-indici (interi), di modo che la funzione ricorsiva lavori solo su indici (cfr esercizio “Sostanze” della lezione del 28/11/2013). Sono sufficienti gli ADT Tabella di Simboli “elementari”, cioè quelli visti sino al 28/11/2013.

**Esercizi ulteriori (non oggetto di valutazione, ma utili ai fini della preparazione all’esame)**

**Esercizio n. 3: quadrato magico**

Si scriva un programma che, letto da tastiera un numero positivo  $N$ , generi e visualizzi a video un quadrato magico di dimensione  $N \times N$ . Un quadrato magico di lato  $N$  è una matrice di interi che contiene tutti e solo i valori compresi nell’intervallo  $[1, N^2]$ , e tale che la somma degli elementi di ogni sua riga, di ogni sua colonna e delle sue due diagonali sia costante (pari a  $N \cdot (N^2 + 1) / 2$ ).

**Esempio:**

Sia  $N$  pari a 3. Un possibile quadrato magico risulta essere il seguente (il valore della somma magica è uguale a  $3 \cdot (9 + 1) / 2 = 15$ ):

2	7	6
9	5	1
4	3	8

**Suggerimento:**

Si noti la somiglianza con l’esempio visto a lezione di generazione delle permutazioni, dove le differenze sono:

- i valori sono noti a priori appartenere all’intervallo  $[1, N^2]$
- i valori sono organizzati su una matrice quadrata e non su un vettore
- non è richiesto di generare tutte le soluzioni, bensì solo la prima (se esiste).

Si può procedere interpretando la specifica come un opportuno problema di generazione delle permutazioni e successivamente visualizzare ognuna delle soluzioni trovate come matrice oppure, in alternativa, operando direttamente sulla matrice.

Non essendo richiesto pruning, la verifica dell’accettabilità della soluzione può essere fatta una volta raggiunto il caso terminale.

**Esercizio n. 4: gioco del Sudoku**

Il gioco del *Sudoku* consiste in una matrice di dimensione  $N^2 \times N^2$  (tipicamente  $N$  vale 2 oppure 3, oppure 4), solo parzialmente piena, nella quale ogni casella contiene un numero tra 1 e  $N^2$ . La matrice va completata in modo che ciascuna riga e ciascuna colonna contenga tutti i numeri tra 1 e  $N^2$ . Inoltre, se si suddivide la matrice in  $N^2$  blocchi di dimensione  $N \times N$ , ogni blocco deve contenere tutti i numeri tra 1 e  $N^2$ .

Scrivere un programma  $C$  che, letta la matrice iniziale da un file, ricerchi una soluzione valida per essa, ovvero compatibile con i valori specificati nel file di ingresso. Il file è costituito da  $N^2$  righe di  $N^2$  numeri ciascuna, separati da uno o più spazi, dove il valore 0 viene utilizzato per indicare una posizione vuota. Si noti che il valore di  $N$  (ovvero, di  $N^2$ ) non è noto a priori, né esso è specificato nel file di ingresso, e deve essere dunque dedotto dal programma stesso in fase di lettura.



**Esempio:**

Supponendo che il file di ingresso abbia contenuto:

```
4 0 0 1
0 1 0 0
3 0 1 0
0 0 0 0
```

il programma deve dedurre che  $N$  vale 2 (matrice  $4 \times 4$ ) e visualizzare una possibile soluzione per tale matrice, come:

```
4 3 2 1
2 1 4 3
3 2 1 4
1 4 3 2
```

**Esercizio n. 5: tour del cavaliere**

Data una scacchiera di dimensione  $N \times N$  (con  $N$  costante predefinita), un cavallo viene posizionato nella casella di coordinate  $(x, y)$ . Supponendo che il cavallo possa muoversi sulla scacchiera seguendo le regole degli scacchi, scrivere un programma in linguaggio C che faccia effettuare al cavallo un insieme di mosse tale da fargli percorrere tutte le caselle della scacchiera una e una sola volta.

Si ricorda che, nel gioco degli scacchi, un cavallo si muove con “balzi” a L, ovvero, data una posizione di partenza  $P$  come in figura, esso può raggiungere con una sola mossa una qualunque delle otto caselle di arrivo denominate  $A$ :

```
.....
..A.A..
.A...A.
...P...
.A...A.
..A.A..
.....
```

**Suggerimento:**

Si utilizzi una matrice di interi per rappresentare la scacchiera. Il valore del generico elemento indichi il numero della mossa che ha portato il cavaliere nella casella stessa. Con una scacchiera di lato 6, ad esempio, partendo dall'angolo in alto a sinistra, una possibile soluzione che si ottiene è la seguente:

```
1 16 7 26 11 14
34 25 12 15 6 27
17 2 33 8 13 10
32 35 24 21 28 5
23 18 3 30 9 20
36 31 22 19 4 29
```