



TECNICHE DI PROGRAMMAZIONE

Esercitazione di Laboratorio 8

Obiettivi

- ☐ Risolvere problemi di verifica/selezione/ordinamento, iterativi, utilizzando vettori e matrici (*Dal problema al programma: Cap. 4 e 5*),

Contenuti tecnici

- ☐ Basi di Input Output
- ☐ Utilizzo di funzioni
- ☐ Costrutti condizionali e iterativi
- ☐ Manipolazioni elementari di vettori e matrici (di int e float)

Da risolvere durante il laboratorio oppure prima/dopo il laboratorio stesso

Esercizio 1.

Competenze: lettura/scrittura di file, manipolazioni di matrici;

Categoria: problemi di verifica e selezione (Dal problema al programma: 4.5) – Problemi complessi (Dal problema al programma: 5)

Individuazione di regioni

Un file di testo contiene una matrice di interi (0 o 1) con il seguente formato:

- la prima riga del file specifica le dimensioni reali della matrice (numero di righe n_r e numero di colonne n_c). Si assuma che entrambi i valori siano al più pari a 50
- ciascuna delle n_r righe successive contiene gli n_c valori corrispondenti a una riga della matrice, separati da uno o più spazi
- ogni cella può contenere solamente il valore 0 (associato al colore bianco) o il valore 1 (associato al colore nero)
- le celle nere sono organizzate in modo da formare regioni rettangolari (ogni regione nera è circondata da una cornice di celle bianche, oppure da bordo/i della matrice). A tal fine, si consideri che l'adiacenza delle celle è considerata solo lungo i quattro punti cardinali principali (Nord, Sud, Ovest, Est), non in diagonale.

Si scriva un programma C che:

- legga la matrice dal file di ingresso (il file non contiene errori, quindi ci sono solo rettangoli neri che rispettano i vincoli)
- individui le regioni nere più grandi per altezza, larghezza e area totale
- in caso di parità, si riporti una tra le regioni individuate che soddisfano un certo criterio
- per ognuna di tali regioni produca in output le coordinate dell'estremo superiore sinistro e le sue caratteristiche (altezza, larghezza, area totale)

Esempio:	Mappa corrispondente:
----------	-----------------------



```
5 6
1 1 0 0 0 0
0 0 1 1 0 0
0 0 1 1 0 1
0 0 0 0 0 1
1 0 1 0 0 1
```

	0	1	2	3	4	5
0						
1						
2						
3						
4						

Output del programma:

Max Base: estr. sup. $SX = \langle 0, 0 \rangle$ $b=2$, $h=1$, Area=2

Max Area: estr. sup. $SX = \langle 1, 2 \rangle$ $b=2$, $h=2$, Area=4

Max Altezza: estr. sup. $SX = \langle 2, 5 \rangle$ $b = 1$, $h = 3$, Area = 3

Esercizio 2.

Competenze: algoritmi di ordinamento iterativi, analisi empirica di complessità

Valutazione di algoritmi di ordinamento

Si considerino i seguenti algoritmi di ordinamento per ordinare in maniera ascendente vettori di interi:

- Selection Sort
- Insertion Sort
- Shell Sort

Si scriva un programma in C che per ogni sequenza numerica acquisita da file (`sort.txt`) invochi tutti gli algoritmi di ordinamento sopra indicati e stampi a video:

- il numero di scambi
- il numero di iterazioni del ciclo esterno
- per ogni passo del ciclo esterno il numero di iterazioni del ciclo interno
- il numero totale di iterazioni.

Il file `sort.txt` è caratterizzato dal seguente formato:

- sulla prima riga appare il numero S di sequenze numeriche
- seguono S righe nella forma $\langle \text{lunghezza} \rangle \langle \text{sequenza} \rangle$ dove $\langle \text{lunghezza} \rangle$ è un intero non negativo (al massimo 100) a rappresentare la lunghezza della sequenza riportata su tale riga, e $\langle \text{sequenza} \rangle$ sono $\langle \text{lunghezza} \rangle$ numeri separati da uno spazio.