

TECNICHE DI PROGRAMMAZIONE, A.A. 2021/2022

Esercitazione di Laboratorio 7

Valutazione: **entrambi gli esercizi 1 e 2** saranno oggetto di valutazione.

Scadenza: caricamento di quanto valutato - entro le 23:59 del 3/6/2022: andranno caricati insieme i laboratori 7 e 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. (Esercizio da consegnare per il bonus-laboratorio)

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. Si assuma che il file non contenga errori (i.e. tutti i rettangoli neri rispettano la formattazione descritta precedentemente)

- individui le regioni nere più grandi, rispettivamente in termini di: (1) altezza, (2) larghezza e (3) area. Nel caso di parità di dimensione, si riporti una tra le regioni che soddisfa il rispettivo criterio.
- per ognuna delle regioni identificate al punto precedente, produca in output le seguenti informazioni: le coordinate (*riga, colonna*) dell'estremo superiore sinistro, l'altezza, la larghezza e l'area.

Esempio:	Mappa corrispondente:
<pre> 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 </pre>	

Output del programma:

Max altezza: estremo=(2,5), altezza=3, larghezza=1, area=3

Max larghezza: estremo=(0,0), altezza=1, larghezza=2, area=2

Max area: estremo=(1,2), altezza=2, larghezza=2, area=4

Esercizio 2. (Esercizio da consegnare per il bonus-laboratorio)

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:

- la prima riga riporta il numero *S* di sequenze numeriche contenute nel file

- le successive S righe riportano, una per riga, le seguenti informazioni:

`<lunghezza> <sequenza>`

dove `<lunghezza>` è un intero non negativo (al massimo 100) che rappresenta la lunghezza della sequenza riportata su tale riga, mentre `<sequenza>` è una sequenza di `<lunghezza>` numeri interi separati da uno spazio.

Esempio del file `sort.txt`:

4

5 1 2 3 4 5

5 1 2 3 4 0

5 5 4 3 2 1

5 1 5 2 4 3