Commento al Laboratorio n. 1

Esercizio n. 1: Giornate di campionato

Il formato del file di ingresso è il seguente:

- la prima riga contiene 2 interi che rappresentano il numero di righe `n` e di colonne `m`
- seguono `n` righe ciascuna delle quali contiene `m` valori interi (0, 1 o 3).

Si assume corretto il file e non si eseguono controlli di errore.

Il `main` legge il file e chiama la funzione `giornate`, che calcola per ogni giornata e per ogni squadra il punteggio di quella squadra in quella giornata e tramite la funzione `cercaMax` individua per ogni giornata la capolista. La complessità della funzione `giornate` (2 cicli `for` annidati) è $\Theta(n \cdot m)$. Il punteggio di ogni squadra è memorizzato nel vettore `punti` di dimensione pari al numero di squadre (`n`). La ricerca del massimo in un vettore di interi comporta ipotizzare un valore iniziale del massimo, scandire il vettore per aggiornare tale valore in funzione di un confronto e ritornare l'indice del valore massimo così trovato. La complessità della funzione `cercaMax` (1 ciclo `for`) è $\Theta(n)$.

Esercizio n. 2: Ricodifica di testo con dizionario

Struttura dati: si assume che sia le parole del testo sia quelle del dizionario non siano più lunghe di una costante `WORD`=50, terminatore incluso. Il dizionario è organizzato come vettore `diz` dimensionato staticamente a `MAXD`=30 voci, definite come `struct` di tipo `entry` con 2 campi `parola` e `token` (stringa di sostituzione).

Decomposizione in sottoproblemi: il problema si decompone naturalmente in 2 sottoproblemi:

- caricamento del dizionario
- lettura del testo riga per riga, con ricerca, sostituzione e scrittura immediata sul file di uscita.

Si evita quindi di immagazzinare il testo in una struttura dati interna, limitandosi quindi a un vettore di caratteri (`riga`).

Lettura del dizionario: la funzione `caricaDizionario` è una banale iterazione di letture da file e memorizzazione nel vettore `diz`, noto il numero di voci del dizionario.

Lettura ed elaborazione del testo: la generica riga viene elaborata percorrendo ognuno dei suoi caratteri e verificando se si tratta dell'inizio di una delle parole nel dizionario (si provano tutte le parole finché non sono terminate oppure si trova una corrispondenza). In caso affermativo, si scrive su file la stringa di sostituzione (token) (e si avanza in riga al termine della parola, altrimenti si scrive direttamente il carattere).

Il confronto tra una parola del dizionario e la sottostringa di riga iniziante all'i-esimo carattere viene proposto in due varianti:

- in una si realizza direttamente (nella funzione `confronta`, che non richiede uso di puntatori) il confronto carattere per carattere tra le stringhe

- in una seconda variante (`confrontaPunt`) si utilizza la funzione di libreria `strncmp`, con la quale è tuttavia necessario identificare la sottostringa di riga mediante aritmetica dei puntatori (`riga+inizio`).

Si noti che, con utilizzo leggermente più avanzato di aritmetica dei puntatori, si sarebbe potuto utilizzare la funzione di libreria `strstr`, che svolge direttamente la ricerca di una sottostringa in una stringa.

Si noti infine che, come scelta alternativa, si sarebbe potuto organizzare la ricerca con un criterio alternativo: per ogni parola nel dizionario, cercane tutte le occorrenze in riga, effettuando le opportune sostituzioni. Pur trattandosi di soluzione equivalente alla precedente, questa richiede di manipolare la stringa in riga, effettuandovi le sostituzioni, in quanto, per ogni parola in dizionario, occorre ricominciare a cercare dall'inizio di riga.

Esercizio n. 3: Rotazione di matrici

Poiché le funzioni ritornano un solo valore, non avendo ancora a disposizione il passaggio di parametri by reference, per ritornare sia il numero di righe, sia il numero di colonne si definisce una struct matmaxN i cui campi sono r e c. La struct è ritornata dalla funzione leggiMatrice.

Si osservi che nell'esempio le righe e le colonne iniziano da 1, anziché da 0 come in C e di questo si tiene conto nel generare gli indici della matrice.

Si presentano 4 soluzioni richiamabili dalla funzione `ruotaMatrice` sulla base di un valore di selezione strategia passato come argomento alla riga di comandi. Le funzioni sono quelle di rotazione di un vettore sviluppate nell'es. 3 del Lab. 0. Si ricorda che la riga di una matrice in C può essere vista come vettore di dimensione pari al numero di colonne. La rotazione per colonne è ricondotta a una rotazione su di un vettore, prima ricopiando la colonna in un vettore, ruotando il vettore e poi ricopiandolo sulla colonna originale.